NASA Technical Memorandum 86145

A General Numerical Analysis Program for the Superconducting Quasiparticle Mixer

Ross G. Hicks, Marc J. Feldman,
and Anthony R. Kerr

JULY 1986

NASA

NASA Technical Memorandum 86145

A General Numerical Analysis Program for the Superconducting Quasiparticle Mixer

Ross G. Hicks, Marc J. Feldman,
and Anthony R. Kerr
NASA Goddard Institute for Space Studies
New York, New York

National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1986

This Page Intentionally Left Blank

TABLE OF CONTENTS

APPENDIX

ABSTRACT

This report describes a user oriented computer program SISCAP (SIS Computer Analysis Program) for analyzing SIS mixers. The program allows arbitrary impedance terminations to be specified at all LO harmonics and sideband frequencies. It is therefore able to treat a much more general class of SIS mixers than the widely used three-frequency analysis, for which the harmonics are assumed to be short-circuited. An additional program, GETCHI, provides the necessary input data to program SISCAP.

The program SISCAP performs a nonlinear analysis to determine the SIS junction voltage waveform produced by the local oscillator. The quantum theory of mixing is used in its most general form, treating the large signal properties of the mixer in the time domain. A small signal linear analysis is then used to find the conversion loss and port impedances. The noise analysis, includes thermal noise from the termination resistances and shot noise from the periodic LO current. Quantum noise is not considered.

Many aspects of the program have been adequately verified and found accurate. However, this project was terminated before the program was extensively employed, and it is possible that errors may occur in various untested parameter ranges.

An example is given, which forms part of some initial investigations into the effect of decreasing the SIS junction capacitance on the performance of a simple SIS mixer circuit. Early indications are that, for this circuit, a high $\omega_p R_N C_J$ product is desirable.

LIST OF SYMBOLS USED IN THIS REPORT

B	bandwidth
β	thermal energy parameter (1/kT)
C_J	SIS parasitic junction capacitance
Δ	energy gap parameter
δ	Kronecker delta function
δI	small signal current flowing in SIS junction
$\delta I'$	small signal current in augmented small signal circuit
δV	small signal voltage across SIS junction
δi_T	thermal noise current
δi_S	shot noise current
$\delta t'$	time increment
δV	voltage increment
e	electronic charge
f	cycle frequency
$\hbar$	Planck's constant divided by 2π
I_{KK}	Kramers-Kronig function
I_{DC}	DC SIS tunnel junction current flowing in absence of LO
I^{NL}	nonlinear SIS large signal current
I^{LIN}	linear large signal current
I_n^{LIN}	nth Fourier harmonic of I^{LIN}
J	SIS small signal response function
k	Boltzmann's constant
L_{ij}	conversion loss to i-th sideband from the j-th sideband
m	index parameter

m	frequency index
M	numerical accuracy parameter
P	convergence parameter for voltage update method
P_O	noise power out
ϕ	barrier phase
Q	numerical accuracy parameter
R_{ID}	identity element resistance for voltage update method
R_N	normal state SIS resistance
τ	SIS normalized time response parameter
t	time
T	SIS physical temperature
T_M	mixer input noise temperature
T_{DSB}	mixer input double sideband noise temperature
T_{SSB}	mixer input single sideband noise temperature
U	large signal SIS time domain response variable
V	SIS tunnel junction voltage
$V^\dagger$	updated SIS tunnel junction voltage
V_n	nth Fourier harmonic of V
V_{N_0}	noise output voltage
V_{LO}	LO source voltage
V_{DC}	DC source voltage
V_{gap}	SIS gap voltage
V_{TOTAL}	total voltage range required of discretized I-V curve
W_{LO}	small signal analysis variable
ω_p	angular LO frequency
ω	Fourier transform variable

ω_m	m-th sideband frequency
χ	SIS junction impulse response
Y	small signal intrinsic admittance matrix
Y'	small signal augmented admittance matrix
Z'	small signal augmented impedance matrix
Z_e	embedding impedance
Z_c	impedance of the SIS junction capacitance C_J
Z_0	transmission line impedance for multiple reflection technique
Z_m	input impedance of the mixer at the m-th sideband
Z'_0	center row of small signal impedance matrix

2. Introduction

As a result of work by Tucker [1] on the quantum theory of mixing, it is now possible to predict the performance of SIS (superconductor-insulator-superconductor) quasi-particle mixers with reasonable accuracy. This theory has been quantitatively verified in its three frequency approximate form [2]. The purpose of this report is to make available a user oriented computer program SISCAP (SIS Computer Analysis Program) for determining the performance of an SIS mixer with arbitrary terminations at the LO (local oscillator) harmonics and harmonic sidebands. The program uses the Tucker theory in its most general form. It is much more versatile than earlier analyses which employed the three-frequency approximation, for which all LO harmonics above the fundamental and sidebands above the upper sideband of the LO fundamental are assumed short-circuited.

The general outline of this report follows that of a similar work by Siegel and Kerr [3], which described a computer program to analyze Schottky diode mixers using classical mixer theory. Reference [3] has proved very valuable in the analysis of Schottky mixers, and it is hoped that this report will serve the same purpose for SIS mixers.

The mixer analysis program SISCAP is subdivided into three parts. The most difficult part of the program is the nonlinear large signal analysis which is used to determine the LO voltage waveform at the junction. This analysis is done iteratively to find a self-consistent solution and requires a significant amount of CPU time. Following the nonlinear analysis, a linear small signal analysis computes the conversion loss between the input (signal) and output (IF) ports as well as the input and output impedance match. Finally, the program determines the equivalent input noise temperature of the SIS mixer. As an additional feature, this computer program performs the three frequency model analysis if desired. The run time for the example listed in Section A.3 is 0.5 mins. on an AMDAHL V6.

Although this report is written explicitly for a mixer using single SIS junctions, the program may be used to analyze any single-particle tunneling device in the quantum regime whose I-V curve is resistive at large voltages. In particular, the program reduces a series array of SIS junctions to its single junction equivalent by assuming the junctions are identical. In any situation however, the DC I-V curve must always be a monotonic function and the user should ensure that experimental measurement errors do not violate this requirement.

This report also describes the program GETCHI, which must be run prior to SISCAP. GETCHI processes an experimental SIS DC I-V curve to provide input data for SISCAP in proper form, and determines the value of various control parameters for SISCAP. Further information on this program is given in Section 3. An empirical study of the digitization constraints required to maintain numerical errors to an acceptably low value is also described.

1.1 SIS Junctions: Definition of Terms

This work is not meant to be a general reference on the theory and operation of SIS mixers. It is assumed that the reader has a solid working knowledge of this topic, and explicit detail is given only where necessary to describe the computer program and the choices made in its design. For general reference, the reader is referred to a recent review paper [4].

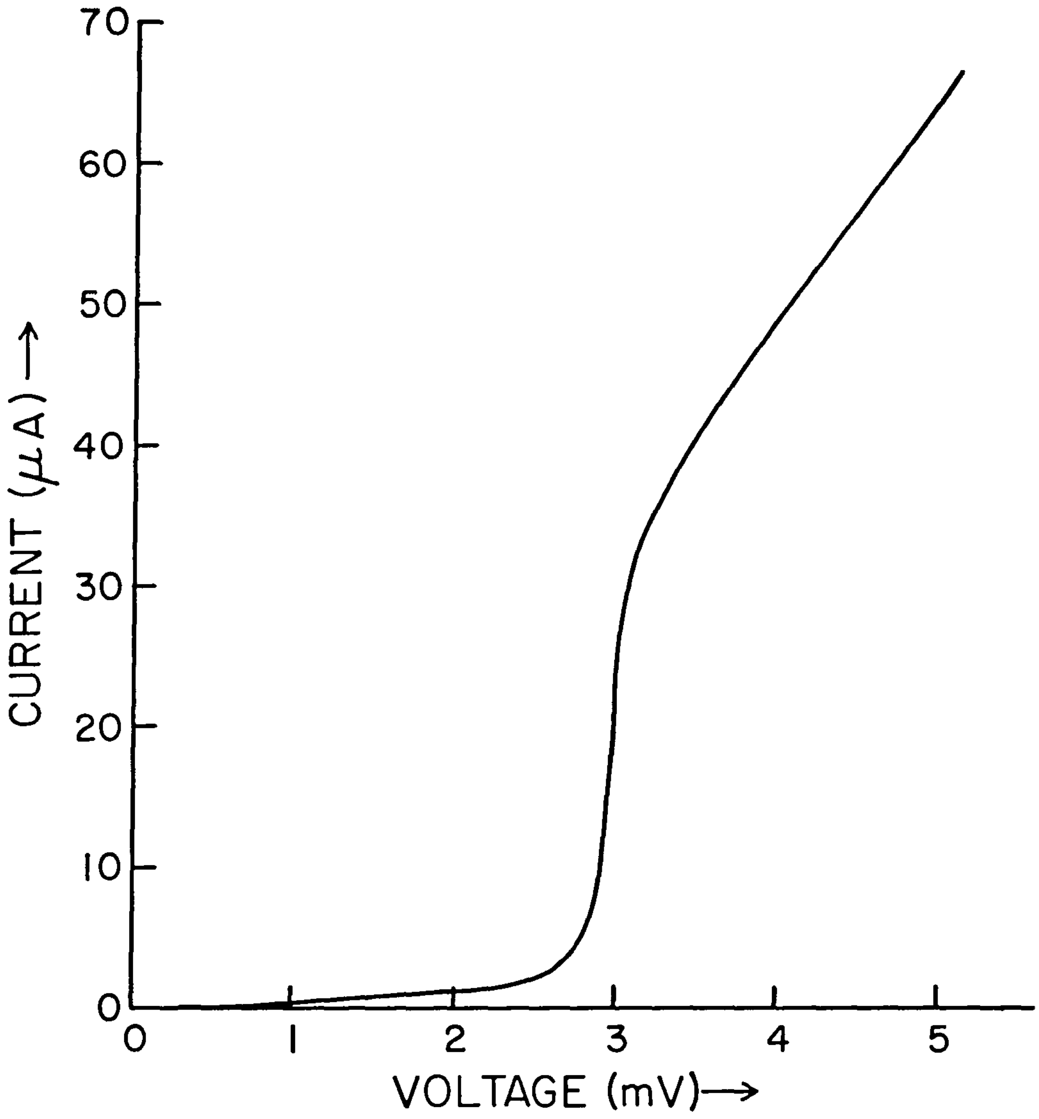

Fig. 1: Typical SIS DC I-V Curve

This section outlines the terms and symbols to be used in this report. The units used here are SI units. A typical quasiparticle DC I-V curve of an SIS junction is shown in Fig. 1. Note that, for this program, superconducting pair tunneling currents are ignored. The rapid rise in current seen in Fig. 1 occurs at the gap voltage V_{gap}, which is related to the energy gap parameter Δ by the formula:

$$V_{gap} = \frac{2\Delta}{e} \qquad (1)$$

At large DC bias voltages, the I-V curve approaches the resistance R_N, which is also the resistance of the junction in the normal state. The SIS junction has the geometrical form of a parallel plate capacitor, whose capacitance C_J acts to shunt the nonlinear quasiparticle current.

In order to describe mathematically the behaviour of the SIS junction, the time domain equations given by Tucker [1] will be used:

$$I^{NL}(t) = \frac{V(t)}{R_N} + \text{Im}\,\{U^*(t) \int_{-\infty}^{t} \chi(t-t')\,U(t')\,dt'\} \qquad (2)$$

$$\chi(t) = \frac{2}{\pi} \int_{0}^{\infty} \left[I_{DC}\left(\frac{\hbar\omega}{e}\right) - \frac{\hbar\omega}{eR_N}\right] \sin\omega t\, d\omega \qquad (3)$$

$$U(t) = \exp\{i\phi(t)\} \qquad (4)$$

$$\phi(t) = -\frac{e}{\hbar} \int_{-\infty}^{t} V(t')dt' \qquad (5)$$

Eq. (1) gives the instantaneous quasiparticle current $I^{NL}(t)$ as a function of the junction voltage V at all time t. The impulse response function $\chi(t)$ of the SIS junction is related to the nonlinear DC I-V curve through the sine Fourier transform as shown in Eq. (3). U(t) represents the driving force applied to the junction and is related via the barrier phase $\phi(t)$ to the instantaneous voltage V(t) in Eqs. (4) and (5).

For the small signal analysis, the SIS junction is treated in the frequency domain. The general frequency domain I-V relation is the Fourier transform of Eq. (2), which is complex. The appropriate small signal response function is then:

$$J(v) = I_{KK}(v) + i\, I_{DC}(v) \tag{6}$$

where $$I_{KK}(v) = P\,\frac{1}{\pi}\int_{-\infty}^{\infty} \frac{\left[I_{DC}(V) - \frac{V}{R_N}\right]}{V - v}\, dV \tag{7}$$

Finally, the problem of analyzing series arrays must be considered. A property of arrays is that their saturation power level increases as N^2, where N is the number of junctions in the array. The conversion gain and noise temperature are independent of N, both in theory [4] and apparently in experiment [5]. Because of these useful properties, the program described in this report has been written to analyze such arrays. The array is reduced to its single junction equivalent on the assumption of identical junctions, by dividing both the current and the voltage of the array DC I-V curve by N [5].

2. Outline of the Necessary Theory

Given the bias voltage, LO amplitude and the DC I-V curve, an SIS mixer analysis should output the following quantities: conversion loss, input impedance, output impedance and mixer noise temperature. These quantities will be calculated by a small signal analysis once the large signal voltage and current waveforms at the SIS junction are known. The first step in the mixer analysis algorithm therefore is to determine these waveforms.

2.1 Large Signal Analysis

The most difficult part of this project is the large signal nonlinear analysis. All of the Fourier coefficients of the LO waveform must be determined in a self-consistent manner. Several papers have appeared in the literature on this problem in the context of the Schottky diode mixer. Early workers such as Torrey and Whitmer [6] assumed the voltage across the diode junction to be sinusoidal, with the implication that the harmonics of the local oscillator were short circuited. Fleri and Cohen [7] removed this assumption but their work was restricted to simple lumped element embedding networks. More realistic embedding networks were analyzed by Egami [8] and by Gwarek [9], using harmonic balance techniques. However, these schemes frequently had difficulty converging for some circuits of practical interest. In particular, the method of Egami [8] tended to diverge when more than three harmonic terms were considered, and the convergence of the method of Gwarek [9] was a strong function of the initial solution estimates.

In a bid to overcome these convergence deficiencies, Kerr [10] developed the multiple reflection technique, in which a hypothetical lossless transmission line of arbitrary characteristic impedance is introduced between the nonlinear diode and the linear circuit elements. This method converges in all cases tested and the convergence rate [11] depends upon how close the harmonic embedding impedances are to the characteristic impedance of the transmission line. There is no dependence upon one's estimate of the initial conditions.

Another technique, developed by Hicks and Khan [11,12], consists of two dual methods, the voltage update and the current update methods. This has also converged for all cases tested, and the rate of convergence depends upon the proximity of the embedding impedances at each LO harmonic to either short circuits or open circuits. In both this technique and the multiple reflection method, the algorithm operates in the time domain when considering the nonlinearity and in the frequency domain when dealing with the linear embedding network.

Most practical SIS mixers have rather large junction capacitances ($\omega_p R_N C_J$[1] products are usually in the range 1 to 10), and so the embedding impedance seen by the nonlinear junction conductance approaches a short circuit with increasing frequency. This condition therefore lends itself to the use of the voltage update method of Hicks and Khan [11] and this method has been adopted for use in the program SISCAP. Such a choice will lead to a rapid convergence rate. Moreover, the use of the voltage update method permits the nonlinear SIS equations to be solved in their simplest form. This is a voltage-input-current-output mode, and requires only a direct calculation from the SIS equations. The multiple reflection technique requires the SIS nonlinear equations to be solved on a

[1] ω_p = angular pump frequency, R_N = normal-state resistance of the SIS device and C_J = junction capacitance.

current-input-voltage-output basis, which adds an extra iteration loop to the process. To improve convergence further, the voltage update program can incorporate a parallel identity resistance technique as outlined in [11]. Typically, identity resistances of the order of $0.5R_N$ have been found to give satisfactory performance.

Finally, for additional flexibility, the multiple reflection method [10] has also been added to this program as an additional option in the large signal analysis section and should be used where low $\omega_p R_N C_J$ products (less than 0.5) are encountered. In such cases, considerable experience is necessary for correct use of the voltage update method and the relatively "hands free" multiple reflection technique is preferable.

2•1.1 Solution of the Large Signal Problem

The equivalent circuit of the mixer is shown in Fig. 2. For analysis of this mixer, the circuit is bisected at the linear-nonlinear interface with each half treated separately. The junction capacitance, C_J, is treated as being part of the linear circuit.

The expressions governing the behaviour of the intrinsic[2] SIS junction are given by Tucker [1] in the time domain as follows:

$$I^{NL}(t) = \frac{V(t)}{R_N} + \mathrm{Im}\,\{U^*(t) \int_{-\infty}^{t} \chi(t-t')\, U(t')\, dt'\} \qquad (2)$$

$$\chi(t) = \frac{2}{\pi} \int_0^{\infty} [I_{DC}(\frac{\hbar\omega}{e}) - \frac{\hbar\omega}{eR_N}]\ \sin\omega t\ d\omega \qquad (3)$$

$$U(t) = \exp\{i\phi(t)\} \qquad (4)$$

[2] The expression intrinsic junction is used in this paper to refer to the nonlinear tunneling admittance of the junction, considered separate from the junction capacitance.

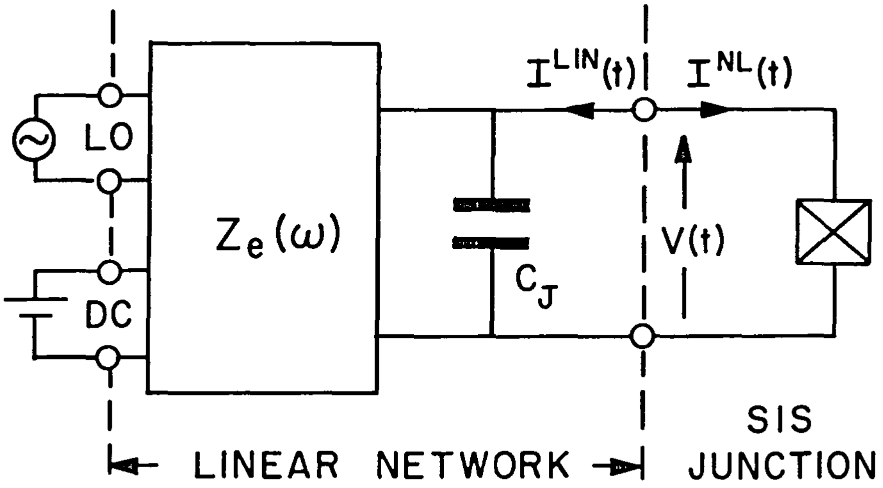

Fig. 2: SIS Mixer Equivalent Circuit

$$\phi(t) = -\frac{e}{\hbar}\int_{-\infty}^{t} V(t')dt' \qquad (5)$$

where $I_{DC}(V)$ is the measured DC I-V characteristic of the SIS junction with no LO power applied;

R_N is the normal resistance of the junction;

$V(t)$ is the instantaneous voltage across the junction; and

$I^{NL}(t)$ is the instantaneous quasiparticle tunneling junction.

The conditions imposed on the steady state response of the mixer circuit by the embedding network can be more conveniently expressed in the frequency domain. Referring to Fig. 2:

$$\frac{V_n}{I_n^{LIN}} = \left[Z_e(n\omega_p) \;//\; Z_c(n\omega_p)\right] \qquad n = 2,3,\ldots. \qquad (8)$$

$$\frac{V_1 - V_{LO}}{I_1^{LIN}} = \left[Z_e(\omega_p) \;//\; Z_c(\omega_p)\right] \qquad (9)$$

$$\frac{V_0 - V_{DC}}{I_0^{LIN}} = \left[Z_e(0)\right] \qquad (10)$$

where V_n and I_n^{LIN} are the amplitudes of the Fourier components of $V(t)$ and $I^{LIN}(t)$ at frequency $n\omega_p$, V_{LO} and V_{DC} are the amplitudes of the Thevenin equivalent LO and DC voltage sources, and $Z_e(n\omega_p)$ is the impedance of the equivalent external embedding circuit at frequency $n\omega_p$, and $Z_c(n\omega_p) = -i/n\omega_p C_J$.

2.1.2 Voltage Update Algorithm

The iterative voltage update algorithm [11] works as follows:

(i) An initial estimate is made of the large signal steady state voltage waveform across the nonlinear tunnel junction, V(t);

(ii) Eqs. (2) - (5) are used to determine the current $I^{NL}(t)$ which is produced by the voltage V(t) across the intrinsic junction;

(iii) Using Kirchhoff's current law, $I^{LIN}(t) = -I^{NL}(t)$;

(iv) A fast Fourier transform gives $I^{LIN}(\omega)$ from $I^{LIN}(t)$;

(v) This current, $I^{LIN}(\omega)$, which flows into the embedding network, must be maintained by a voltage $V^{\dagger}(\omega)$ whose various Fourier components are given by Eqs. (8) - (10);

(vi) Using an inverse fast Fourier transform, $V^{\dagger}(t)$ is obtained from $V^{\dagger}(\omega)$;

(vii) $V^{\dagger}(t)$ and V(t) are compared; if "equal", the iteration is complete. If not, a new V(t), equal to $pV^{\dagger}(t) + (1-p)\ V(t)$, is defined, and the cycle repeated from step (ii). The convergence parameter, p, is normally a value in the range $0 < p \leqslant 1$ [11].

This procedure is called the "voltage update method", because the mechanism for approaching the self-consistent solution in a controllable fashion is the updating of the junction voltage waveform with each iteration in step (vii). It may be shown [11] that this method converges when, at all the harmonics of interest, the magnitude of the impedance of the linear embedding network is smaller than the magnitude of the impedance of the nonlinear circuit. This is the case in an SIS mixer by virtue of the junction capacitance, considered here as part of the embedding network. The value of the parameter, p, is chosen on the basis of previous experience [11]. Essentially, its value represents a tradeoff between speed of convergence and the risk of divergence. In this work, p was fixed at unity since the use of an identity element (to be discussed below) proves sufficient to guarantee convergence.

The computer program allows parallel resistive identity elements [11] to be inserted at the linear-nonlinear interface. A resistive identity element consists of the parallel combination of a resistance R_{ID} and a resistance $-R_{ID}$. The net effect of the two parallel resistances on the circuit performance is zero since one cancels the effect of the other. However, in the large signal analysis, convergence is improved since the resistance R_{ID} is lumped in parallel with the linear embedding network and in a similar manner, the resistance $-R_{ID}$ is added in parallel with the nonlinear circuit [11]. The effect of such an addition is to enhance the voltage update properties of the circuit by increasing the effective impedance of the tunnel junction and simultaneously decreasing the input impedance of the embedding network. From experience, setting R_{ID} equal to $0.5R_N$ has been found effective.

2.1.3 Multiple Reflection Technique

The multiple reflection algorithm [10] provides an alternative to the voltage update technique discussed in the previous section. In this method, a lossless transmission line of characteristic impedance, Z_0, is inserted at the linear-nonlinear interface. Its length is an integral number of LO wavelengths. The steady state operation of the modified network is identical to that of the original circuit, but the approach to steady state differs from that of the original circuit. Also, because the solution approaches steady state in a physically realizable way, the multiple reflection method should not be susceptible to numerical instability.

The iterative procedure proceeds as follows:

(i) A right propagating wave from a source impedance Z_0 is emitted from the embedding network and impinges on the SIS device. From the nonlinear circuit equations, the periodic voltage and current produced by the incident wave at the SIS device terminals may be calculated.

(ii) The nonlinear SIS junction generates a left propagating wave containing, in general, all harmonics of the LO. After a delay, the left propagating wave impinges on the embedding network.

(iii) From the reflection coefficient of the embedding network, a new right propagating wave may be deduced.

This iterative cycle of steps (i) - (iii) continues until convergence is reached. At convergence, the SIS junction voltage waveform remains unchanged with increasing iteration number. This method should converge for any value of Z_0, the transmission line characteristic impedance, although the rate of convergence is affected. Typically, Z_0 equal to R_N has been found to give good results.

2.2 Small Signal Analysis

The small signal analysis follows precisely the theory as formulated by Held and Kerr [13] for the analysis of Schottky diode mixers, with the conversion admittance matrix elements given by Tucker [1]. The small signal and noise theory given here is based largely on reference [14].

Using the results of the large signal analysis together with a knowledge of the embedding impedances at the various sideband frequencies, a linear small signal conversion admittance matrix may be formulated for the mixer. This matrix gives the relationships between the small signal sideband currents and voltages at the nonlinear element, and hence gives all the small signal properties of the mixer.

2.2.1 Frequency and Subscript Notation

If a mixer is pumped at frequency ω_p and has an intermediate frequency ω_0, then, under the linear assumption, the only small signals which can produce an IF response are at the sideband frequencies $\omega_0 + m\omega_p$, $m = 0, \pm 1, \pm 2, \ldots$. Following Saleh [15] it is useful to define the sideband frequencies by:

$$\omega_m = \omega_0 + m\omega_p \qquad m = 0, \pm 1, \pm 2, \pm 3, \ldots \quad (11)$$

It may be seen that lower sideband frequencies are represented by negative terms ($m < 0$). A brief comment on the meaning of these negative frequency terms is given in the footnote.[3]

Saleh's frequency notation leads to a considerable simplification of the mathematics of mixer theory. Using this notation all upper sideband frequencies ($\omega_0 + |m|\omega_p$) are considered positive, while all lower sideband frequencies ($\omega_0 - |m|\omega_p$) are negative. The sideband frequency index m is

[3] Electrical quantities are frequently described by a single complex quantity associated with some frequency, assumed positive. For example, a voltage of frequency ω may be described simply by its complex half-amplitude V, implying an instantaneous voltage $v(t) = V e^{j\omega t} + V^* e^{-j\omega t}$. It is just as meaningful to work with a negative frequency ($-\omega$) and the conjugate of the complex half-amplitude (V^*), provided the convention is clearly understood. Impedances and admittances are then simply the conjugates of their conventional positive frequency values, i.e. $Z(-\omega) = V^*/I^* = Z^*(\omega)$.

used as a subscript to the various electrical quantities and hence the upper sideband, intermediate, and lower sideband frequencies are: $\omega_{+1} = \omega_0 + \omega_p$, ω_0, and $\omega_{-1} = \omega_0 - \omega_p$; and V_{+1}, V_0 and V_{-1} represent voltages at these frequencies.

2.2.2 Small Signal Conversion Matrix

Using the sideband notation described in the previous section, let $\hat{\delta I}$ and $\hat{\delta V}$ denote the vectors of the small signal sideband currents (δI_n) and voltages (δV_n) at the terminals of the intrinsic SIS junction. Then

$$\hat{\delta I} = [\ldots, \delta I_1, \delta I_0, \delta I_{-1}, \ldots]^T \qquad (12)$$

and

$$\hat{\delta V} = [\ldots, \delta V_1, \delta V_0, \delta V_{-1}, \ldots]^T \ . \qquad (13)$$

Torrey and Whitmer [6] have shown that $\hat{\delta I}$ and $\hat{\delta V}$ are related via a conversion admittance matrix $\hat{Y}$ defined by

$$\hat{\delta I} = \hat{Y}\,\hat{\delta V} \ . \qquad (14)$$

If the row and column numbering of the square matrix $\tilde{Y}$ correspond with the sideband numbering, $\hat{Y}$ can be written out as:

$$\tilde{Y} = \begin{array}{ccccccc} & & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & & \\ \cdot & \cdot & Y_{11} & Y_{10} & Y_{1-1} & \cdot & \cdot \\ \cdot & \cdot & Y_{01} & Y_{00} & Y_{0-1} & \cdot & \cdot \\ \cdot & \cdot & Y_{-11} & Y_{-10} & Y_{-1-1} & \cdot & \cdot \\ & & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & & \end{array} \qquad (15)$$

with element values given by [1]:

$$\begin{aligned} Y_{mm'} = (-i) \frac{e}{2h\omega_{m'}} \sum_{n,n'=-\infty}^{\infty} W_{LO}(n\omega_p) \overset{*}{W}_{LO}(n'\omega_p)\, \delta_{m-m',n'-n} \\ \{[J(n\omega_p + eV_0/\hbar) - J(n\omega_p - \omega_{m'} + eV_0/\hbar)] \\ - [J^*(n'\omega_p + \omega_{m'} + eV_0/\hbar) - J^*(n'\omega_p + eV_0/\hbar)]\} \end{aligned} \qquad (16)$$

where W_{LO} is defined by:

$$\exp\{-i \frac{e}{\hbar} \int_{-\infty}^{t} [V(t') - V_0]\, dt'\} = \sum_{n=-\infty}^{\infty} W_{LO}(n\omega_p) e^{-in\omega_p t} \quad ; \qquad (17)$$

V(t) is the instantaneous large signal voltage across the tunnel junction;

δ is the Kronecker delta function;

J is defined in Section 2.2.5;

V_0 is the DC component of the junction waveform; and

$*$ denotes the complex conjugate operation.

The matrix $\hat{Y}$ can be regarded as the admittance matrix of a multifrequency multiport network, as shown in Fig. 3, in which there is one port for every sideband frequency ω_m. If the parallel combination of the embedding impedances and the parasitic junction capacitive impedance, $Z_{e_m} // Z_{c_m}$, corresponding to the sideband frequencies, ω_m, are now connected in parallel with the intrinsic junction, an augmented network is formed as shown by the broken line in Fig. 3. The ports of the augmented network correspond to the terminals of the intrinsic SIS junction at the various sideband frequencies and do not represent physically accessible ports in the real mixer. The augmented network can be described by the admittance matrix $\hat{Y}'$, defined by:

$$\hat{\delta I}' = \hat{Y}' \, \hat{\delta V} \tag{18}$$

where

$$\hat{\delta I}' = [\ldots, \delta I'_1 , \delta I'_0 , \delta I'_{-1} , \ldots]^T$$

and

$$\tilde{\delta V} = [\ldots, \delta V_1, \delta V_0, \delta V_{-1}, \ldots]^T , \tag{19}$$

δV_m and $\delta I'_m$ are the small signal voltage and current components, at sideband $\omega_m = \omega_0 + m\omega_p$ (port m) of the augmented network. The elements of the augmented admittance matrix $\tilde{Y}'$ are given by:

$$Y'_{mn} = Y_{mn} \qquad m \neq n \tag{20a}$$

and

$$Y'_{mm} = Y_{mm} + [Z_{e_m} // Z_{c_m}]^{-1} , \; m = n \tag{20b}$$

Inverting (18) gives

$$\tilde{\delta V} = \tilde{Z}' \, \tilde{\delta I}' , \tag{21}$$

where

$$\tilde{Z}' = (\tilde{Y}')^{-1} . \tag{22}$$

The impedance matrix $\tilde{Z}'$ enables us to calculate the conversion loss and the input and output impedances of the mixer and is also needed in computing the noise temperature.

2•2•3 Mixer Port Impedances

The impedance Z_m of any port of the intrinsic SIS junction (see Fig. 3) can be found by open circuiting the corresponding embedding impedance Z_e and then forming the $\widehat{Z}'$ matrix defined by (21). The desired port impedance is given by the mm-th element of the newly formed $\widehat{Z}'$ matrix:

$$Z_m = Z'_{mm,\infty} \quad , \tag{23}$$

where the subscript ∞ indicates that $\widehat{Z}'$ has been formed with Z_{e_m} open circuited. In such an operation, the corresponding mixer input impedance seen by the embedding circuit therefore is:

$$Z_{in_m} = Z_m = Z'_{mm,\infty} \quad . \tag{24}$$

In particular, the IF output impedance is given by:

$$Z_{IF_{out}} = Z_{in_0} = Z_0 = Z'_{00,\infty} \quad . \tag{25}$$

2.2.4 Conversion Loss

The conversion loss from sideband j to sideband i in a mixer is

$$L_{ij} \overset{\Delta}{=} \left(\frac{\text{power available from the signal source at sideband } \omega_j}{\text{converted power from the signal source, delivered to the load at sideband } \omega_i}\right) .$$

Consider for the moment only the intrinsic SIS junction shown in Fig. 3. The power available from impedance $(Z_{e_j} \ // \ Z_{c_j})$ at the j-th sideband is

$$P_{available} = 1/4 \ |\delta I'_j|^2 \ \mathrm{Re}[Z_{e_i} \ // \ Z_{c_j}] \quad . \tag{26}$$

The power delivered to a load impedance $(Z_{e_i} \ // \ Z_{c_i})$ at sideband i is, using (21):

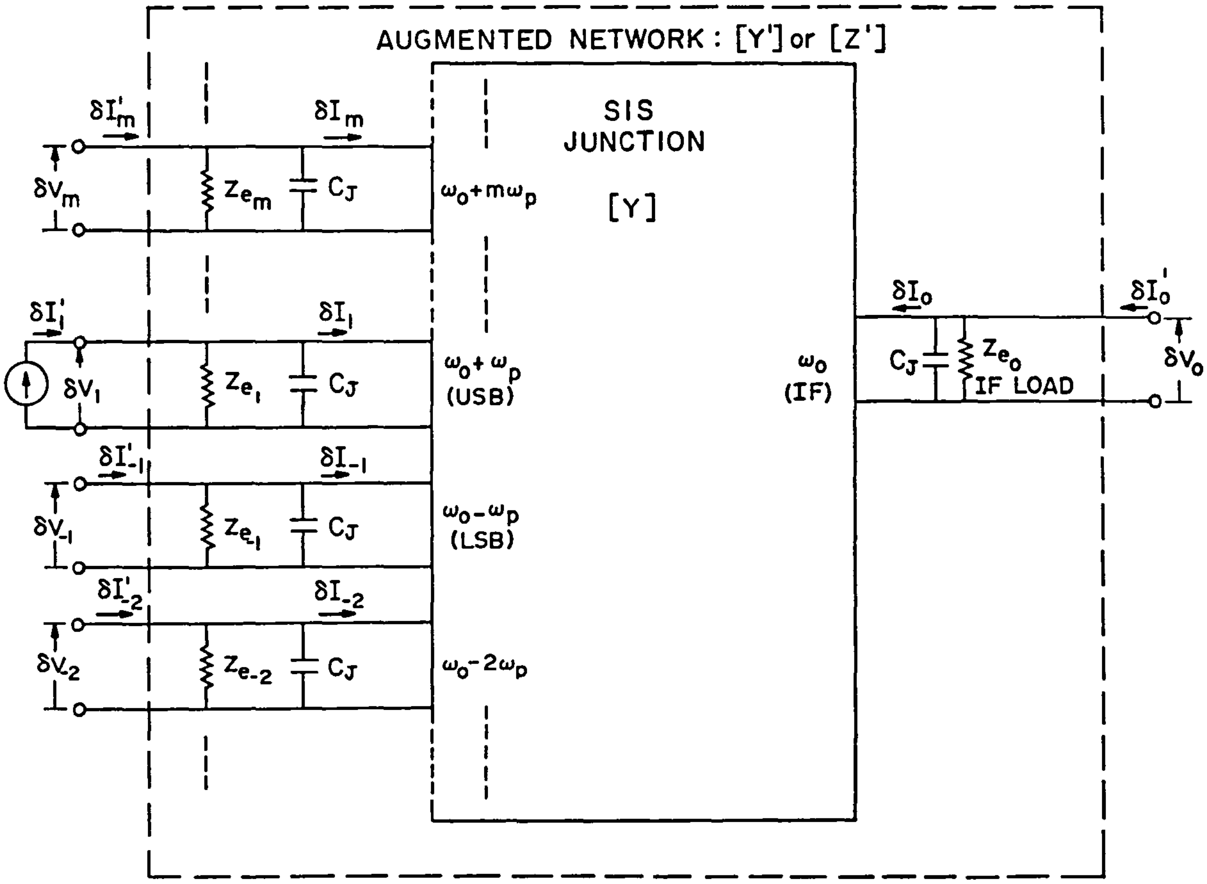

Fig. 3: Multifrequency Multiport Equivalent Small Signal Circuit

$$P_{delivered} = \frac{|Z'_{ij}|^2 \; |\delta I'_j|^2 \; \mathrm{Re}[Z_{e_i} \; // \; Z_{c_i}]}{|Z_{e_i} \; // \; Z_{c_i}|^2} \tag{27}$$

Dividing (26) by (27) gives the conversion loss L'_{ij} of the intrinsic tunnel junction:

$$\frac{P_{available}}{P_{delivered}} = L'_{ij} = \frac{|Z_{e_i} \; // \; Z_{c_i}|^2 \; |Z_{e_j} \; // \; Z_{c_j}|^2}{4|Z'_{ij}|^2 \; \mathrm{Re}[Z_{e_j} \; // \; Z_{c_j}] \; \mathrm{Re}[Z_{e_i} \; // \; Z_{c_i}]} \tag{28}$$

The conversion loss of the actual mixer, L_{ij}, is equal to L'_{ij} since the linear parasitic junction capacitance dissipates no power.

2•2•5 Calculation of J

The value of the function J is required in the calculation of the small signal conversion parameters. J is a complex function of a real argument, defined as:

$$J(v) = I_{KK}(v) + i\, I_{DC}(v) \tag{6}$$

where $i = \sqrt{-1}$, $I_{DC}(v)$ is the current in the SIS junction at DC voltage v, and I_{KK} is the Kramers-Kronig transform of I_{DC}, defined mathematically as follows:

$$I_{KK}(v) = P\,\frac{1}{\pi}\int_{-\infty}^{\infty} \frac{\left[I_{DC}(V) - \frac{V}{R_N}\right]}{V - v}\, dV \qquad (7)$$

The following properties of Eq. (7) are worth noting:

(i) The Kramers-Kronig transform is identical to the Hilbert transform [16].

(ii) Note that Eq. (7) contains a singularity at $V = v$. To enable a finite integration to be calculated, the Cauchy principal value is used wherein the singularity is approached equally from both directions on the V axis.

(iii) The ohmic portion of the I-V characteristic is subtracted out of the numerator of the integrand in the above integral such that the integral is finite. Fig. 4 shows the area under the I-V curve is infinite unless the linear or ohmic portion is removed. The linear or ohmic portion of the DC I-V curve contributes only a constant term to the value of I_{KK}. As only subtractions of I_{KK} terms are relevant to the calculation of the value of the mixer properties, the linear portion of the DC I-V curve may be subtracted out without error [17].

(iv) The calculation of the Kramers-Kronig transform does not proceed directly according to the definition given above in Eq. (7), but via the Fourier transform as follows [16]:

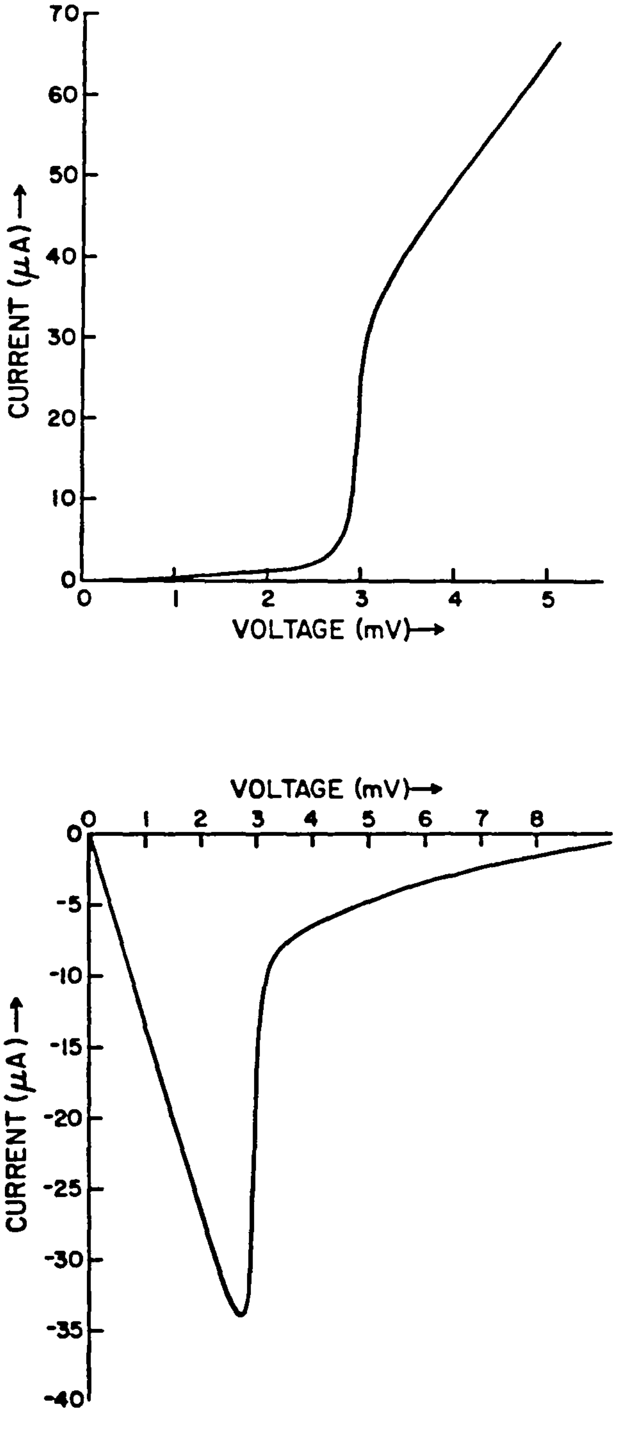

Fig. 4: (a) SIS DC I-V Curve; (b) Same I-V Curve With Ohmic Portion Removed.

(i) Form the Fourier transform of the function $[I_{DC} - \frac{V}{R_N}]$;

(ii) For values in the positive Fourier domain, multiply by -i; for values in the negative Fourier domain, multiply by +i;

(iii) Invert the resulting transform using a fast Fourier transform.

This method represents a significant saving in CPU time, particularly in view of the need for fast Fourier transform routines in other parts of the computer programs GETCHI and SISCAP.

2•3 Mixer Noise Theory

The noise observed in a SIS tunnel junction comes from three sources: (i) shot noise due to the statistical nature of the current flow across the junction, (ii) thermal noise due to the random motion of the charge carriers in the embedding network, and (iii) quantum noise, which occurs by virtue of the zero point energy.

A theory capable of predicting the contribution of the quantum noise in a circuit analysis computer program has yet to be derived. Indeed, the problem of quantum noise in tunnel junction mixers, discussed in Refs. [4] and [18], has not been clearly resolved, and it will be ignored here. General considerations [19] require that a "high gain linear amplifier", such as the SIS mixer, add _at least_ a half photon of fluctuation energy, referred to its input, to any incoming signal. This results in a _minimum_ noise temperature $T_m = \hbar\omega/2k$, which is small at the frequencies of interest (2.8K at 115GHz).

The equivalent circuit of the SIS tunnel junction, including noise sources, is shown in Fig. 5. $\overline{\delta i_T^2}$ and $\overline{\delta i_S^2}$ are the mean-square values of the thermal and shot noise currents in the frequency range f to f + Δf. These current sources can be regarded as generating a multitude of quasi-sinusoidal frequency components, each with its own amplitude and phase. In the multifrequency multiport equivalent circuit of the mixer (Fig. 3), the noise sources can be included by connecting a noise current source at each sideband frequency to the appropriate port of the augmented network.

2•3•1 Thermal Noise

Thermal noise generated in any embedding resistance has components which are uncorrelated at the various sideband frequencies. Let $\delta I'_{T_m}$ represent the quasi-sinusoidal component at sideband frequency ω_m of the thermal noise current source in Fig. 5 and let δV_{T_m} be the sideband noise voltage produced by $\delta I'_{T_m}$. The noise voltage produced at the IF port of the augmented network (Fig. 3) by the thermal noise at all the sidebands can be found using Eq. (21):

$$\delta V_{T_0} = \hat{Z}'_0 \; \hat{\delta I}'_T \;, \tag{29}$$

where $\hat{\delta I}'_T = [\ldots, \delta I'_{T_1}, \delta I'_{T_0}, \delta I'_{T_{-1}}, \ldots]$ is the vector of input thermal noise currents at the sideband ports of Fig. 3 and $\hat{Z}'_0$ is the center row of the augmented impedance matrix $\hat{Z}'$. From (29):

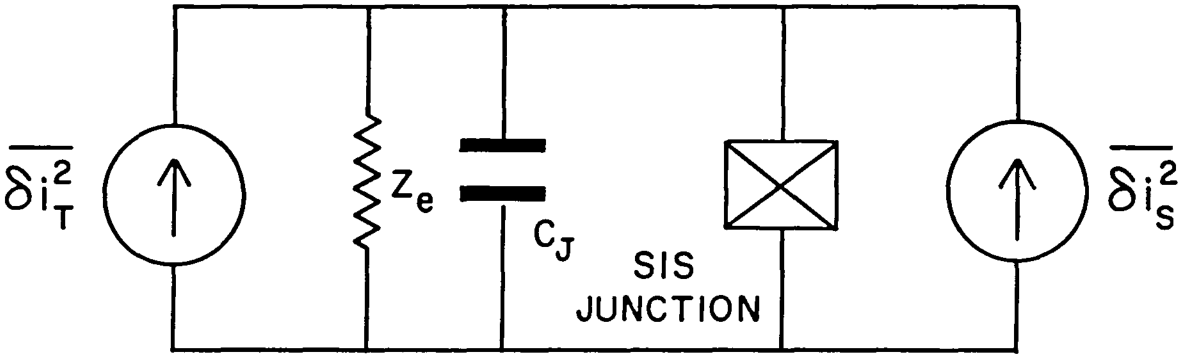

Fig. 5: SIS Noise Equivalent Circuit

$$\delta V_{T_0} \cdot \delta V^*_{T_0} = \hat{Z}'_0\, \hat{\delta I}'_T \cdot \left(\hat{Z}'_0\, \hat{\delta I}'_T\right)^\dagger = \hat{Z}'_0\, \hat{\delta I}'_T\, \hat{\delta I}'^\dagger_T\, \hat{Z}'^\dagger_0 \; . \qquad (30)$$

where † implies the complex conjugate transpose of the vector. Taking the ensemble average gives:

$$\left\langle \left| \delta V_{T_0} \right|^2 \right\rangle = \hat{Z}'_0 \left\langle \hat{\delta I}'_T\, \hat{\delta I}'^\dagger_T \right\rangle \hat{Z}'^\dagger_0 \; . \qquad (31)$$

The square matrix $\langle \hat{\delta I}'_T\, \hat{\delta I}'^\dagger_T \rangle$ is the thermal noise current correlation matrix. Since the thermal noise components at the various sideband frequencies are uncorrelated, the matrix is diagonal. Including the Planck quantum correction to the Rayleigh-Jeans blackbody formula, this matrix is:

$$\langle \delta I'_{T_m}\ \delta I'^\dagger_{T_{m'}} \rangle = 0 \qquad\qquad m \neq m' \qquad (32a)$$

$$\langle \delta I'_{T_m}\ \delta I'^\dagger_{T_m} \rangle = \frac{4\hbar\omega_m B \quad \mathrm{Re}(1/Z_{e_m})}{\left[\exp(\hbar\omega_m/kT) - 1\right]} \qquad \text{for } m = m' \qquad (32b)$$

2·3·2 Shot Noise

The shot noise in a mixer arises from the current produced in the tunnel junction conductance by the local oscillator and DC bias. The formal derivation for the output shot noise follow the thermal noise derivation. The mean square shot noise voltage produced at the IF port of the mixer is given by an equation analogous to Eq. (31) but with subscript T replaced by subscript S.

The shot noise can be considered as white (Gaussian) noise, amplitude modulated by the LO waveform. Dragone [20] and Uhlir [21] have investigated the properties of this modulated noise and have shown that there is a partial correlation between the quasi-sinusoidal components at the various sideband frequencies. The correlated components at these sidebands are down converted in the diode to the intermediate frequency where they add vectorially. Therefore, the shot noise current correlation matrix is <u>not</u> diagonal. It is given by Tucker [1] for quantum mixer theory as follows:

$$\langle \delta I'_{S_m} \; \delta I'^{\dagger}_{S_{m'}} \rangle = e \sum_{n,n'=-\infty}^{\infty} W_{LO}(n\omega) \; W^{*}_{LO}(n'\omega) \; \delta_{m-m',n'-n}$$

$$\{\coth\left[\beta(eV_0 + n'\hbar\omega + \hbar\omega_{m'})/2\right] \; I_{DC}(V_0 + n'\hbar\omega/e + \hbar\omega_{m'}/e)$$

$$+ \coth\left[\beta(eV_0 + n\hbar\omega - \hbar\omega_{m'})/2\right] \; I_{DC}(V_0 + n\hbar\omega/e - \hbar\omega_{m'}/e)\} \quad (33)$$

where $\beta = 1/kT$.

2•3•3 Total Mixer Noise

The total output noise of the mixer is obtained by combining the thermal and shot noise components:

$$\langle|V_{N_0}|^2\rangle = \hat{Z}'_0 \; [\langle\hat{\delta I}'_S \; \hat{\delta I}'^{\dagger}_S\rangle + \langle\hat{\delta I}'_T \; \hat{\delta I}'^{\dagger}_T\rangle] \; \hat{Z}'^{\dagger}_0 \quad (34)$$

The noise power delivered to the IF load from the mixer is:

$$P_0 = \langle|V_{N_0}|^2\rangle \; \mathrm{Re}[Z_{e_0}]/|Z_{e_0}|^2 \quad (35)$$

The equivalent input noise temperature T_M of the mixer is the temperature to which the signal source conductance must be heated to give the same output noise from a noiseless but otherwise identical mixer as the actual mixer would produce when its signal source conductance was maintained at absolute zero temperature. Thus,

$$T_M \overset{\Delta}{=} P_0 \, L_{01}/kB \quad . \tag{36}$$

Note that Eq. (36) does not use the quantum corrected form of Johnson noise. The quantum corrected form used as an alternate definition would give a slightly different numerical result. In either case the noise <u>power</u> is unambiguous.

T_M, as defined above, is the single sideband noise temperature, T_{SSB}. When describing the performance of a mixer whose physical input port is coupled to both the signal and image frequencies, it is more convenient to talk in terms of a double sideband noise temperature T_{DSB}. T_{DSB} is the temperature to which the signal and image source conductances must be heated to give the same output noise from a noiseless but otherwise identical mixer as the actual mixer would produce when its signal and image source conductances were maintained at absolute zero temperature. For mixers in which the conversion loss from the signal and image frequencies, L_{01} and L_{0-1}, are equal:

$$T_{DSB} = T_{SSB}/2 \tag{37}$$

and in general when $L_{01} \neq L_{0-1}$

$$T_{DSB} = \frac{T_{SSB}}{1 + \dfrac{L_{01}}{L_{0-1}}} \tag{38}$$

2•4 Comment on the Effect of Various Program Control Parameters

In transforming these procedures into a workable computer program, there is a practical limit on the number of harmonics of the local oscillator which can be computed in the nonlinear large signal analysis. Also, the small signal admittance matrix $\hat{Y}$ will be truncated above some finite sideband number.

For the voltage update large signal analysis, this truncation is equivalent to terminating the intrinsic junction in a short circuit at all frequencies higher than the truncation frequency. For a large enough number of harmonics, this approximation is reasonable since the $\omega_p R_N C_J$ product of the junction is usually greater than unity. These same remarks apply to the truncation of the small signal admittance conversion matrix $\hat{Y}$.

For the multiple reflection large signal analysis, the justification for the truncation is not so clear. In this method, the truncation is equivalent to terminating the intrinsic SIS junction in the characteristic impedance of the transmission line at higher frequencies. Although the parasitic capacitance clearly leads to short circuit terminations at these higher harmonics, it has been found in practice that the errors so caused by the truncation are not significant.

It should be noted that, in spite of the high $\omega_p R_N C_J$ products, a large number of points (e.g. 64 harmonics of 113.9GHz LO) still needs to be considered by the program since the intrinsic response of the junction is still exceptionally fast (e.g. 0.5 psec rise time for 3mV gap voltage device). However, embedding impedance measurements do not have to be made right up to the 64th harmonic, since the shorting effect of the junction capacitance say above the 10th harmonic makes the mount impedance external to the capacitance irrelevant; the junction merely sees a short circuit. The number of harmonics treated in the program is automatically equal to one half of the number of time intervals into which each LO cycle is divided (Sec. 3.4. v.i).

2•5 Nonlinear SIS Device Equation Calculations

Three problems arise in putting the nonlinear quasiparticle current-voltage equation,

$$I^{NL}(t) = \frac{V(t)}{R_N} + \text{Im}\,\{U^*(t)\int_{-\infty}^{t} \chi(t-t')U(t')dt'\} \qquad (39)$$

in a form suitable for numerical solution. These are:

(i) How to approximate the lower limit of $-\infty$ in the integration;

(ii) How small must the increment $\delta t'$ be ($\delta t'$ is the numerical approximation to dt'); and

(iii) As the two terms on the right hand side of the above equation are approximately equal in magnitude and opposite in sign for all DC bias voltages below the energy gap voltage, relative errors in I^{NL} are likely to be large in this region.

The problem of the lower limit of $-\infty$ is identical to the problem of deciding how much of χ is required before its asymptotically decreasing contributions at higher arguments are insignificant. The solution to this problem is obtained by numerically evaluating $\int_0^M \chi(t)\, dt$ and $\int_0^{2M} \chi(t)\, dt$. M is changed until the two integrals agree with each other to within 0.1%. This procedure was done for both an ideal SIS junction and also a typical real SIS junction. The results of this investigation are as follows:

$$M_{ideal} \geqslant 95\ \tau \tag{40}$$

$$M_{real} \geqslant 16\ \tau \tag{41}$$

where $\tau = \frac{h}{2\Delta}$ is a normalizing time constant, with h = Planck's constant and Δ = the energy gap. It is emphasized that the value of M_{real} will depend on the sharpness of the DC I-V curve of the device to be analyzed. The curve so used to obtain the value of 16τ was relatively sharp as may be seen in Fig. 6. Should a user have a significantly sharper DC I-V curve, then by trial and error, the value of the program control parameter NOCHI should be varied to obtain the appropriate value.

It should be noted at this stage that the integral to be evaluated is $\int_{-\infty}^{t} \chi(t-t')\ U(t')\ dt'$ and not $\int_0^{\infty} \chi(t')\ dt'$ as used above. By experiment, it has been found that selecting a lower limit based on an 0.1% error in the latter integral leads to no more than 0.1% error in the former.

The second problem, the size of δt, is solved in the following manner. The following two numerical integrals, expressed as sums, are computed:

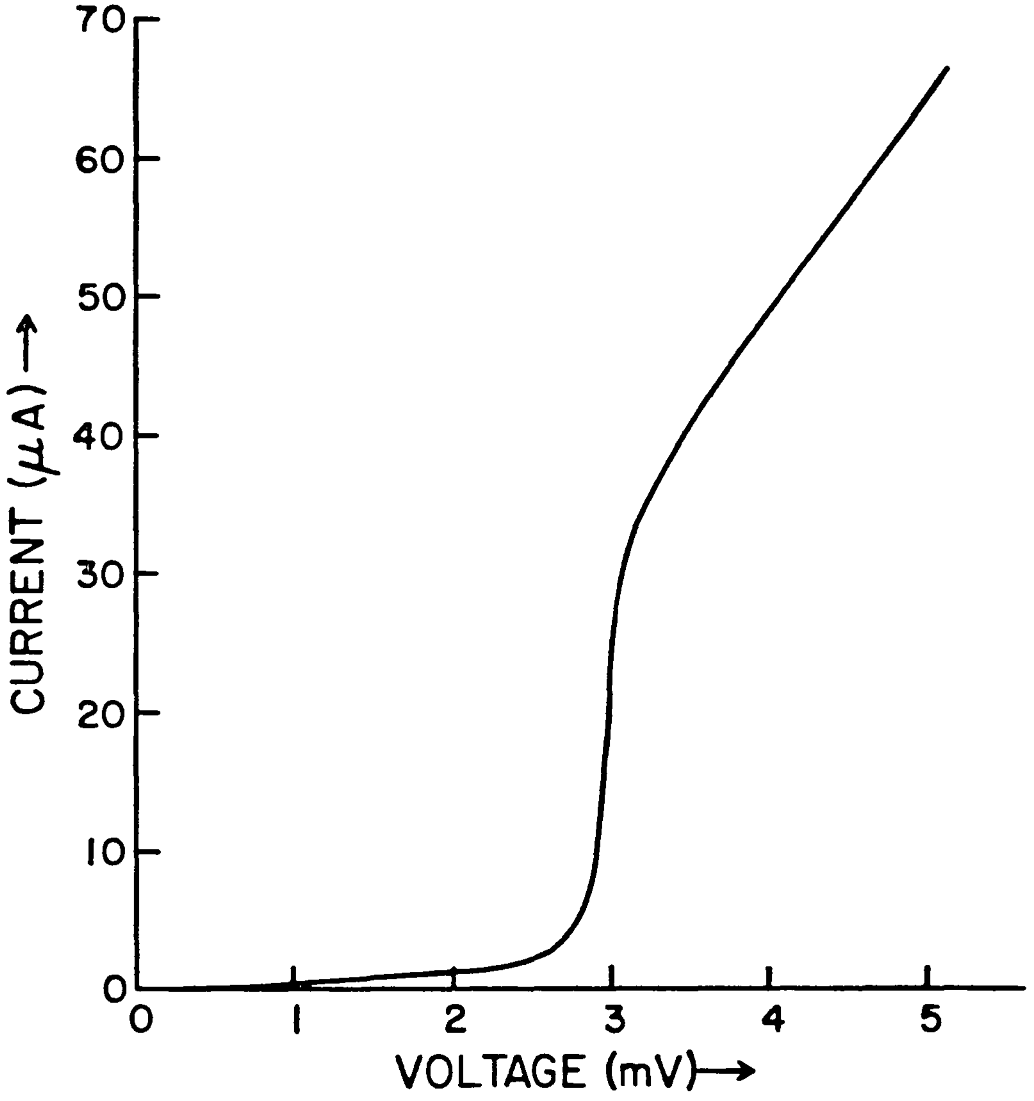

Fig. 6: I-V Curve Used to Determine Numerical Constraint Parameters in Section 2.5

$$\sum_{k=1}^{N} \chi(k\delta t)\delta t \qquad \text{as} \qquad \sum_{k=1}^{2N} \chi(\frac{k\delta t}{2}) \frac{\delta t}{2}$$

δt is adjusted until the two sums are within 0.1% of each other. The value obtained for δt is the same for both real and ideal SIS DC I-V curves, since the rise time of the χ function in each case is the same. The value of δt obtained is:

$$\frac{\delta t}{\tau} < 0.064 \tag{42}$$

for both real and ideal SIS DC I-v curves.

The third problem with the nonlinear SIS circuit equation lies in the addition of the two nearly equal and opposite terms. These terms, for voltages below the energy gap, are approximately equal in magnitude but opposite in sign. This is so since the sum must equal the leakage current in the case of zero LO power applied. Even with LO power applied, the DC pumped current is often of the same order of magnitude as the leakage current.

The following table indicates the accuracy required in calculating the term $Q = \mathrm{Im}\{U^*(t)\int_{-\infty}^{t} \chi(t-t')\ U(t')dt'\}$ to ensure a 1% error bound in $I^{NL}(t)$ for the SIS junction of Fig. 6:

VOLTAGE	% MAX ERROR IN Q
1mV	0.0062
2mV	0.024
3mV	0.66
4mV	0.86
5mV	0.92
6mV	0.94

TABLE 1

As expected, the error requirements on Q are most severe in the areas below the energy gap if a constant percentage error is to be maintained in $I^{NL}(t)$. However, it would seem intuitively correct that we could relax the percentage error in $I^{NL}(t)$ for very low values of $I^{NL}(t)$ without appreciably affecting the calculated conversion loss or noise temperature of the mixer.

A summary of the digitization constraints on χ based on this discussion is given in Table 2, for a real and the ideal SIS I-V curves. For comparison, the values required to perform an analogous time-domain analysis of the ideal Josephson pair tunneling currents, from Gayley [22], are also presented.

	IDEAL SIS	REAL SIS	IDEAL JOSEPHSON[22]
M	95τ	16τ	80τ
δt	0.064τ	0.064τ	0.020τ
No. of Points of χ	1484	250	4000

TABLE 2

It should be noted that the χ function derived from the SIS real I-V curve decays much more rapidly than that χ function corresponding to the SIS ideal I-V curve (16τ compared to 95τ). Fig. 7 illustrates this. The decay rates for the Josephson pair current χ function and the ideal quasiparticle (SIS) χ function may be shown analytically to be the same and the two independent results obtained for M in Table 2 verify this. The much smaller step δt required for the Josephson pair current case compared to that for the SIS calculations is a result of the shorter response time for the Josephson currents. Typically, the Josephson response time is approximately three times smaller than that for the quasiparticle currents. As may be inferred from the "number of points" values in the table, a substantial saving in computer memory is obtained by analyzing mixers containing real SIS junctions rather than ideal SIS junctions.

Having established the digitization constraints on χ, those for the SIS I-V curve automatically follow. $\chi(t)$ is a sine Fourier transform of the I-V curve (minus the ohmic part), from which it follows that the digitization constraints for the I-V curve are:

$$V_{TOTAL} \geqslant 8\ V_{gap} \tag{43}$$

$$\delta V \leqslant \frac{V_{gap}}{30} \tag{44}$$

where: the DC I-V curve is digitized from 0 to V_{TOTAL} volts using an interval of δV;

$V_{gap} = \frac{2\Delta}{e}$; and

Δ = energy gap value in Joules.

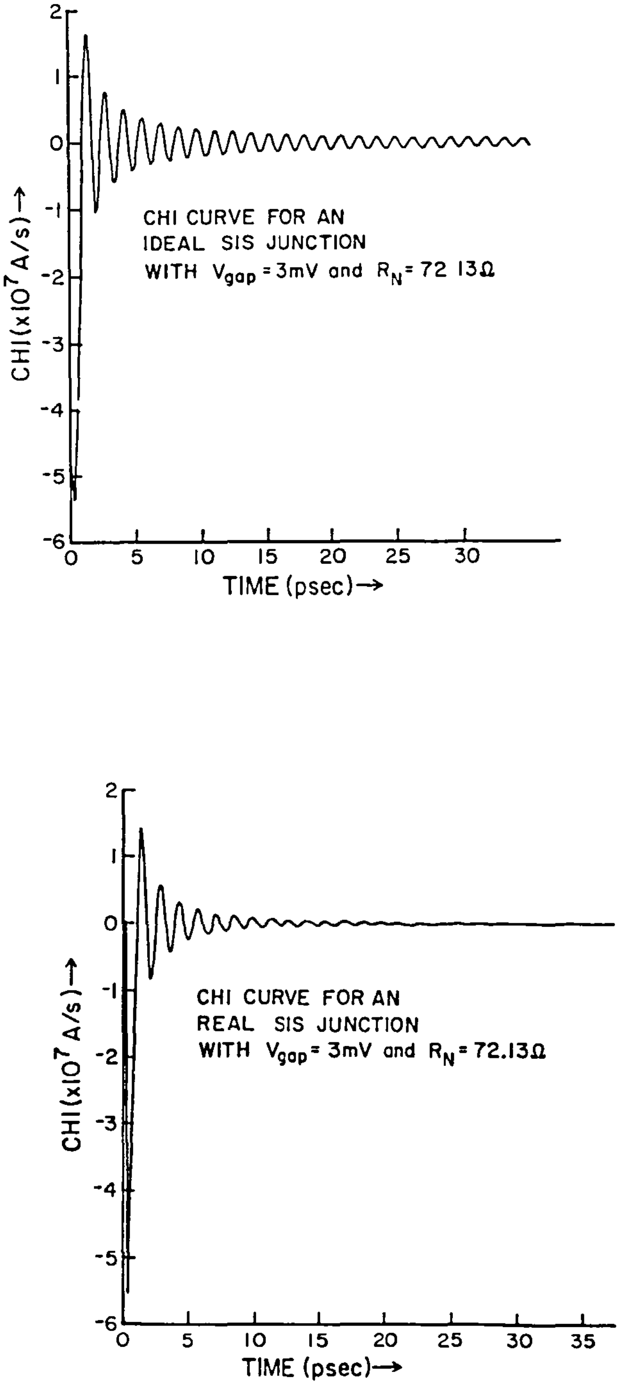

Fig. 7: Comparison Between CHI Function Derived From: (a) An Ideal SIS DC I-V Curve; (b) A Real SIS DC I-V Curve

It should be emphasized that these digitization constraints were obtained from one particular DC I-V curve, albeit a sharp I-V curve. Clearly, different I-V curves will lead to differing digitization constraints. However, on the basis of our experience, the above digitization constraints have ensured sufficient numerical accuracy with all the measured I-V curves we have tested.

The above digitization constraints are embodied into the program GETCHI which is described in Sections 3.1 and 3.2.

3. Description of the Mixer Analysis Program SISCAP and Program GETCHI

Using the theory given in the previous sections, a user oriented computer program was written to analyze the performance of SIS mixers. The programming language used was FORTRAN. The program consists of two parts, SISCAP and GETCHI.

The program GETCHI must be run first, since it provides information on the storage requirements necessary for an accurate analysis in the program SISCAP. Also, GETCHI outputs the discretized I-V curve and the discretized CHI curve in a form suitable for input to the program SISCAP. GETCHI requires the following as inputs: (i) the I-V curve in some arbitrary yet accurate digitized form; (ii) the number of junctions represented in the input DC I-V curve; (iii) the LO frequency; and (iv) the gap voltage of the single equivalent device. Further details on the input/output data structures can be found in Section 3.2.

The program SISCAP requires as inputs the following: (i) the embedding impedances seen by the SIS tunnel junction (excluding the parasitic junction capacitance contribution) at each harmonic of the local oscillator and at the sidebands; (ii) the SIS DC I-V curve (minus the ohmic part, as provided by the program GETCHI); (iii) the SIS χ curve (which is derived from the I-V curve by the program GETCHI, to be described later); (iv) the various operating conditions for the tunnel junction mixer, e.g. DC bias voltage, LO amplitude, the frequencies of the LO, RF and IF; (v) the normal resistance R_N and junction capacitance C_J of the SIS tunnel junction, and finally (vi) the parameters used to control the various numerical calculations within the program.

The output includes (i) the conversion loss, (ii) the RF input impedance; (iii) the IF output impedance, and (iv) the single sideband mixer noise temperature.

The remainder of this section explains both programs in detail and illustrates the steps used for running them. A complete documented listing of the two programs appears in Appendix A and a general flowchart is given in Fig. 8.

3.1 GETCHI Program Implementation

The main routine of program GETCHI calls various subroutines described below in the sequence indicated.

Subroutine EVEN provides an evenly-spaced (along the voltage axis) internal version of the input I-V data which is used as the reference data for subsequent processing by the program. Linear interpolation is used to calculate these points. If the input data is for an array, subroutine EVEN reduces this to its single junction equivalent. Subroutine EVEN requests as input the number of tunnel junctions arrayed. If not an array, unity should be entered. The given I-V data must be input on device 7.

Subroutine REFINE then takes the I-V internal working data produced by subroutine EVEN and extracts the ohmic or linear portion of the I-V curve.

Subroutine INEXP then interpolates the internal working I-V data stored within the program such that the new spacing and number of points correspond to the correct spacing required for further processing prior to the conversion to CHI data. The LO frequency (FREQ) and the single junction gap voltage (VGAP) are required as input so the program can provide lower limits for the number of LO cycles (NOCYC) and number of points per cycle (NOPNTS). These limits ensure that an accurate

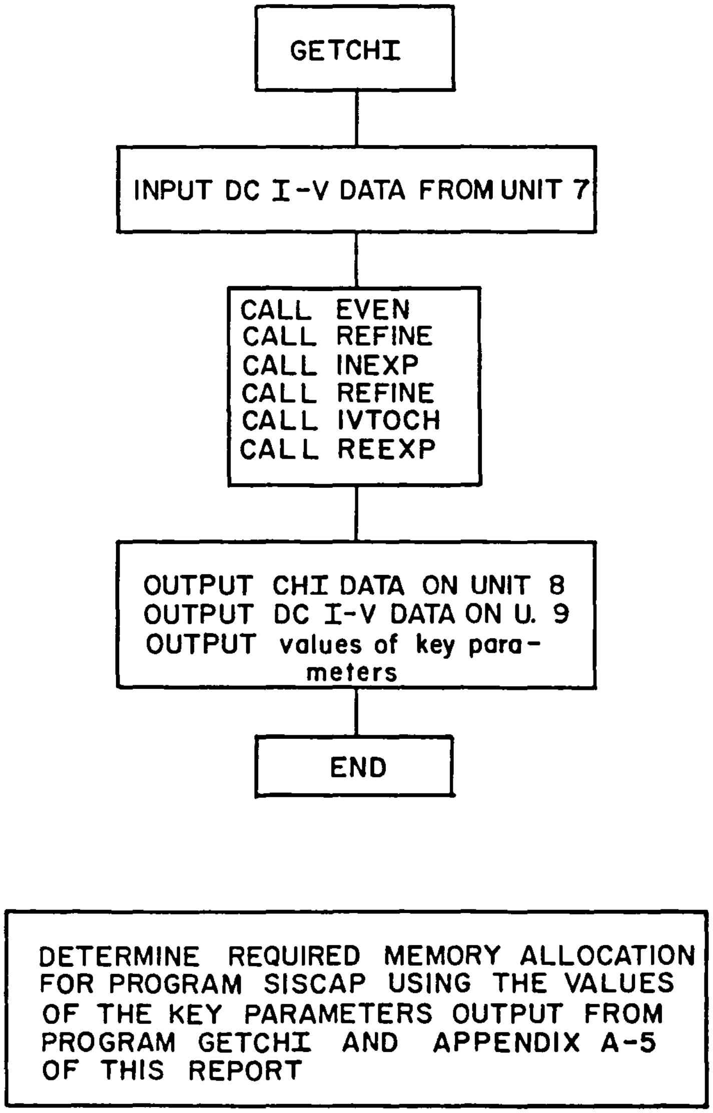
GETCHI
INPUT DC I-V DATA FROM UNIT 7
CALL EVEN
CALL REFINE
CALL INEXP
CALL REFINE
CALL IVTOCH
CALL REEXP
OUTPUT CHI DATA ON UNIT 8
OUTPUT DC I-V DATA ON U. 9
OUTPUT values of key parameters
END
DETERMINE REQUIRED MEMORY ALLOCATION FOR PROGRAM SISCAP USING THE VALUES OF THE KEY PARAMETERS OUTPUT FROM PROGRAM GETCHI AND APPENDIX A-5 OF THIS REPORT

Fig. 8(a): Flowchart of Program GETCHI

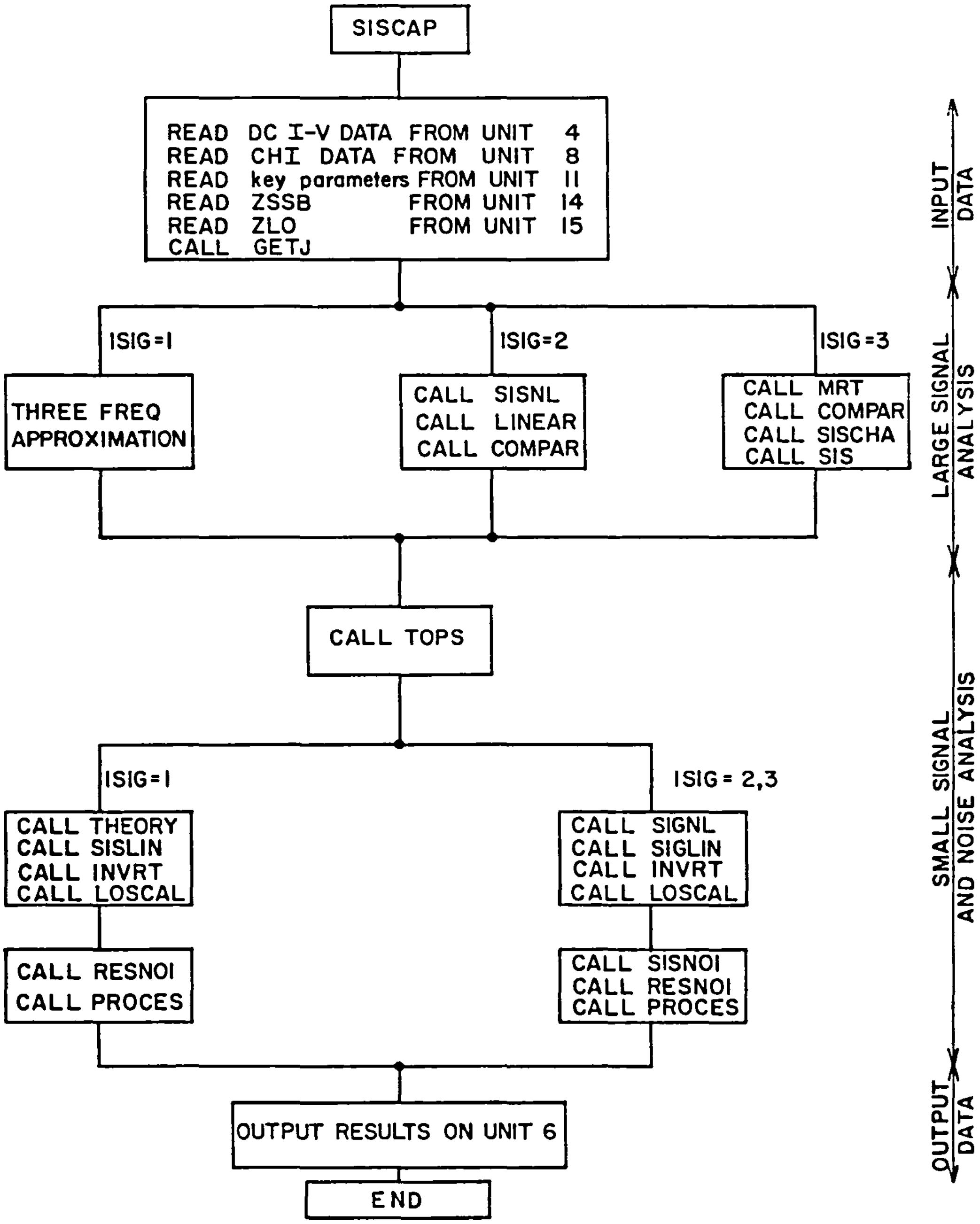

Fig. 8(b): Flowchart of Program SISCAP

calculation is performed in the program SISCAP.

Following this, subroutine REFINE is again called to ensure that the last point of the I-V data is zero current.

Subroutine IVTOCH uses a fast Fourier transform and converts the working I-V data to the equivalent CHI data file for output on device unit number 8.

Finally, subroutine REEXP outputs suitably digitized I-V data for use in evaluating the current I_{DC} and the complex valued parameter J in the SISCAP program. To output this I-V data, the program requests both the number of points required (NOIV) and the spacing between the points (DELTAV). Suggested values for these parameters are NOIV = 1024 and DELTAV = 0.02 mV for a 3mV gap SIS device. For other gap voltage values, DELTAV should be linearly scaled to suit. The DC I-V data is output on device unit number 9.

Output to the console from program GETCHI is a summary of the values of key parameters to be input to program SISCAP. The parameters are: NOCHI, NOIV, NOPNTS and RN. The first three parameters are also used to determine the memory allocation requirement in SISCAP as per Appendix A.5.

3.2 Running the GETCHI Program

A listing of the program GETCHI appears in Appendix A along with the output of a run. The comments in the listing provide a description of the FORTRAN coding.

Prior to running program SISCAP, GETCHI must be run to provide information on memory storage allocation for SISCAP, values of key parameters in program SISCAP as well as to provide the CHI and' the DC I-V curve, each with the appropriate discretization for input to program SISCAP. GETCHI requires as input to it, the DC I-V curve of the SIS junction. Multiple junction I-V curves may be input to GETCHI but the two output files from GETCHI containing CHI and the DC I-V data will reflect those values appropriate to the equivalent single junction device. The program will request the number of junctions (NOJUN) in the series array; a value of unity should be input if a single junction is used.

The number and distribution of the points given in the input DC I-V data file to program GETCHI are at the discretion of the user but must conform to the following guidelines:

(i) the first I-V data point must be the origin (0.0,0.0);

(ii) the input I-V points need not be evenly spaced;

(iii) the span of the input points must be such as to extend well into the linear portion of the I-V curve, i.e. the points must adequately cover the entire nonlinear portion (typically to a voltage four times the gap voltage); and

(iv) the I-V points must represent a monotonic function.

In order to obtain the CHI function, it is necessary to input the frequency of the local oscillator (FREQ), the gap voltage of the single junction equivalent SIS device (VGAP), the number of points per LO cycle (NOPNTS) to be used in the calculation and the number of cycles of points required to accurately describe CHI (NOCYC) in accordance with the conclusions of section 2.5. The program offers assistance in deciding what the minimum acceptable values for NOPNTS and NOCYC are. Both these numbers must be powers of 2. The parameter, NOCHI, whose value must be known for input to the program SISCAP, is equal to NOPNTS*NOCYC. As such, the value of NOCHI is always greater than or equal to NOPNTS. The program also outputs the value of RN to be input to the program SISCAP. The program requests both the number of points for output (NOIV) to the file for program SISCAP and the voltage spacing between these points (DELTAV). The number of points (NOIV) must be a power of 2 and 1024 has been found to give satisfactory results. DELTAV, the voltage spacing between the points, is typically 0.02 mV for a 3mV gap device. The DC voltage span covered by these typical parameters is 20.48 mV (= 1024 x 0.02mV) which clearly covers the nonlinear portion of the I-V curve.

3.3 SISCAP Program Implementation

The program commences in the MAIN routine and inputs all the parameters necessary to control the operation of the program. Next, the values of CHI (χ) and the SIS DC I-V characteristic are read in. At this point, control is transferred to the subroutine GETJ in order to obtain the Kramers-Kronig transform of the DC I-V curve and hence the complex function J. Control is then transferred back to the MAIN routine to input the embedding impedances at the local oscillator harmonics (subroutine GETZ) and sideband frequencies (subroutine GETSSB).

All information has now been input and the program is now ready to perform the large signal analysis. There are three methods which may be used:

(i) the voltage update method for $\omega_p R_N C_J$ products in excess of 0.5;

(ii) the multiple reflection technique for other $\omega_p R_N C_J$ products; and

(iii) the three frequency approximate analysis.

The voltage update method proceeds as follows. An estimate of the junction voltage is made. A call is made to subroutine SISNL which calculates the current flowing into the SIS device, given the input voltage. Next, subroutine LINEAR calculates the voltage required across the embedding network terminals consistent with the current calculated in subroutine SISNL. Finally, COMPAR performs a comparison between the new voltage waveform and the initial estimate of the voltage waveform to determine if a solution has been obtained.

Alternatively, the multiple reflection technique may be used. This proceeds by a call to subroutine MRT which performs the overall iterative algorithm. It in turn calls subroutine SISCHA which sets up various vectors such that the succeeding call to subroutine SIS will solve the nonlinear SIS tunnel junction equations.

If required, a three frequency model calculation can be performed instead of the more general nonlinear analysis routines described above.

Following the completion of the large signal analysis calculations by whatever large signal analysis method, a call to TOPS initiates calculations to obtain the small signal and noise results. TOPS is the principal subprogram controlling the signal and noise analysis programs. It initializes the working matrices to zero and calls SIGNL and SIGLIN. SIGNL calculates the matrix conversion elements which are computed from the SIS device parameters together with the large signal analysis waveforms. The embedding impedances at all the relevant sidebands are calculated by subroutine SIGLIN and, by virtue of the linearity, the matrix so calculated is diagonal. The combination of the subroutines SIGLIN and SIGNL produce the augmented admittance matrix. The combination of the two subroutines INVRT and LOSCAL invert the augmented admittance matrix and perform the arithmetic to calculate the conversion loss and the output impedance. The noise analysis follows next using calls to SISNOI and RESNOI. SISNOI computes the contribution from the shot noise of the SIS device while the thermal or Johnson noise is computed from subroutine RESNOI. Subroutine PROCES then takes the complete noise matrix and, using the previously calculated impedance conversion matrix, computes the equivalent input noise temperature of the mixer.

If a three frequency analysis is invoked, the small signal and noise analyses are calculated as follows. Subroutine THEORY provides the 3x3 matrix elements describing the small signal conversion and noise correlation process. Subsequent calls to subroutines SIGLIN, INVRT and LOSCAL calculate the conversion loss of the mixer. The single-sideband mixer noise temperature is calculated by calls to subroutines RESNOI and PROCES.

For a more detailed description of the operation of the program, the reader is referred to the comments in the program listing and the general flow chart of Fig. 8.

3·4 Running the Mixer Analysis Program SISCAP

A listing of the SIS mixer analysis program SISCAP appears in Appendix A along with the output of a run. Using the IBM Fortran 77 compiler, the execution time for this particular listing is 0.5 mins on an AMDAHL V6 computer. The comments in the listing provide a description of the FORTRAN coding. To run the program, the following information must be supplied as files accessed via the following FORTRAN unit numbers:

(i) UNIT NO. 15: The embedding network impedances at the LO frequency and the higher harmonics. Preceding the impedance values, there must be a number indicating the number of impedances in the file. It is read in under I5 format. All impedance values are to be in ohms and are read in as complex variables using format 2E15.8. The program expects there to be NOPNTS/2 impedances in this file. If there are less than this, the program assumes that those impedances given are zero. It is important to note that these impedance values should not include the effect of the parasitic junction capacitance.

(ii) UNIT NO. 14: The sideband impedances in complex form, again in ohms. The ordering of information in this file is as follows: number of impedances in the file, IF impedance, fundamental USB impedance $(Z_e(\omega_1))$, fundamental LSB impedance $(Z_e(\omega_{-1}))$, second LO harmonic USB impedance $(Z_e(\omega_2))$ and the second LO harmonic LSB impedance $(Z_e(\omega_{-2}))$, etc. Lower sideband impedances should not be input in their complex conjugate form, the program does this operation. As for the previou; case, a number indicating the number of impedances in the file must be submitted as the first entry in the data file, using the format I5. The complex impedances are read in under 2E15.8 format. The program expects there to be (NOPNTS - 1) impedances in the files. If there are less than this, the program assumes that those impedances not given are zero. It is important to note that these impedance values should not include the effect of the parasitic junction capacitance.

(iii) UNIT NO. 8: The DC SIS I-V curve is input in the form of the number of points (format I5) in the file followed by the digitized I-V curve (format 2E15.8). The ohmic part, R_N, will have been subtracted out by program GETCHI (See Section 3.2) which should be used to provide the DC I-V curve data.

(iv) UNIT NO. 4: The CHI values for the SIS tunnel junction device with the following format: the number of points in the file (I5), followed by the values of CHI (2E15.8). The points for this file can be generated by using the program GETCHI (See Section 3.2).

(v) UNIT NO. 11: This file contains all the SIS device parameters together with the various program option values which affect the operating and performance characteristics of program. The parameters and their function are listed sequentially below (all integers are input in the format I5, all real values are input in the format E15.8):

(a) FREQ: frequency of the local oscillator (LO) in Hertz;

(b) FIF: intermediate frequency (IF) in Hertz;

(c) CAP: parasitic junction capacitance of the SIS device in Farads;

(d) RN: normal resistance of the SIS device in ohms. This value is supplied by the program GETCHI;

(e) TEMP: operating temperature of the junction in Kelvins;

(f) IBASE2: binary logarithm of NOIV;

(g) NOIV: number of points in the I-V data file (this must be a power of 2). The value of NOIV must be ᐟhe value used in the program GETCHI. Assistance in selecting the value of NOIV is given when program GETCHI is run (See Section 3.2);

(h) IBASE1: binary logarithm of NOPNTS;

(i) NOPNTS: number of points into which each LO cycle is subdivided (this must be a power of 2). This value must be obtained from the program GETCHI.

(j) MMAX: number of sidebands to be considered (e.g. 9 implies 4 upper, 4 lower and the IF sidebands are tυ be included in the analysis). MMAX must be an odd number. Typically, a value of 15 has been'found to give satisfactory results. Its value is always limited to a maximum of

NOPNTS/2 - 1;

(k) NMAX: number of photon points to be considered in the small signal analysis. This establishes the number of terms summed in Eq. 16 to calculate the admittance matrix elements. At typical LO drive levels, at least 6 photon points on each side of the DC bias point are required to ensure reasonable accuracy. Therefore, the minimum for NMAX is 2*6 + 1 = 13. The maximum value for NMAX is NOPNTS/2 - 1;

(l) ISIG: indicates the type of mixer analysis to be performed: 1 = three port model, 2 = general mixer analysis using voltage update method, 3 = general mixer analysis using multiple reflection technique;

(m) NOPER: maximum number of iterations allowed in the large signal general mixer analysis. This parameter acts as an emergency brake in case the nonlinear analysis does not converge. Typically, its value should be 100. The value of this parameter is irrelevant if the three port approximation is used (ISIG = 1);

(n) NOCHI: number of points required in the CHI (χ) file. The value of this parameter is obtained from the program GETCHI;

(o) NODAT: number of different DC bias voltage cases to be analyzed (see also DELV and VSTART);

(p) TOL: the maximum relative error allowed between successive iterations in the large signal analysis before the solution has converged. Since the computational accuracy of each computer varies from machine to machine, the value assigned to TOL must therefore reflect the respective rounding error of each individual machine. Typically, TOL equal to 0.00001 has been found to be effective on IBM 32 bit machines;

(q) EPS: in a vector of values, EPS is the fraction of the maximum component below which all values are considered to be insignificant and therefore unduly affected by rounding error and the like. Since the computational accuracy of each computer varies from machine to machine, the value assigned to EPS must therefore reflect the respective rounding error of each individual machine. Typically, EPS equal to 0.00001 has been found to be effective on IBM 32 bit machines;

(r) DELV: increment in DC bias voltage between the different bias cases considered for analysis (in volts). See also NODAT and VSTART;

(s) VSTART: first of the NODAT values of DC bias voltage (in volts). See also NODAT and DELV;

(t) VLOSTR: first of the local oscillator peak voltage values (in volts). See also DELVLO and NLO;

(u) DELVLO: increment in local oscillator peak voltage value (in volts). See also VLOSTR and NLO;

(v) NLO: number of different LO voltage cases to be analyzed. See also VLOSTR and NLO;

(w) P: convergence parameter used in the voltage update large signal analysis; its value lies in the range $0 < P \leq 1$. Typically, its value is unity;

(x) RID: value of the identity resistance (in ohms) to be used in the voltage update large signal analysis method. Typically, R_{ID} set equal to $R_N/2$ has been found to give satisfactory results;

(y) ZO: value of the characteristic impedance (in ohms) of the transmission line used in the multiple reflection large signal analysis method. Typically, Z_0 set equal to R_N has been found to give satisfactory results.

Output is received from program SISCAP as follows:

(a) UNIT NO. 6: the console output.

3.5 Verification Tests

The program has been verified for the case of large junction capacitance by comparison with the predictions of the three-frequency analysis. In a number of cases (not restricted to a large junction capacitance), the large signal waveforms predicted by this program were found to be consistent with those calculated by an independently developed SIS nonlinear analysis program which was used to study chaos [23]. The mixer temperatures predicted by this program have not been independently verified but do appear reasonable. This project was terminated before the program was extensively employed, and it is possible that errors may occur in various untested parameter ranges.

4. Example

In this section, the mixer analysis program is used to study the effects of parasitic junction capacitance on the performance of an SIS mixer. The experimentally measured I-V curve, shown in Fig. 9, of an SIS element [24] for which $R_N = 72\ \Omega$ was assumed to be in a mixer circuit with a 113.9 GHz local oscillator. The intermediate frequency was 1.0 MHz and the IF load resistance was 50 Ω. In this low IF limit the signal, image, and LO source impedances must all be equal, and they were chosen to be 55 + i92 Ω (which includes C_J), the value which maximizes the conversion efficiency at a dc voltage in the center of the first photon step in the three-frequency approximation. This SIS mixer was analyzed for a wide range of junction capacitance. In each case the termination impedance at all higher harmonics and sidebands (not including the LO and its sidebands) was the parallel combination of the capacitive reactance and a resistance of 72 Ω (arbitrarily chosen equal to R_N).

For each value of capacitance the LO power and the dc voltage were optimized for maximum conversion on the first photon step. Typical junction current and voltage waveforms resulting from the numerical computation in Section A.3 are shown in Fig. 10. The overall results for varying junction capacitances are not plotted, for fear that relatively large interpolation errors may have made them unreliable. The trend of the results is however quite clear.

The largest value of mixer conversion gain, a few dB, was found in the limit of large capacitance, and agreed with the three-frequency approximation. The conversion gain decreased to a minimum of about unity at an $\omega_p R_N C_J$ product of unity, but then increased again for smaller $\omega_p R_N C_J$ values. The mixer noise temperatures corresponding to these results showed an even less dramatic variation with capacitance, remaining between 20 and 25 K. The mixer output impedance was negative at all capacitance values, while the input impedance was always positive. It must be emphasized that even if these conclusions are verified for this specific mixer circuit, they may be far from typical and should not be taken as general.

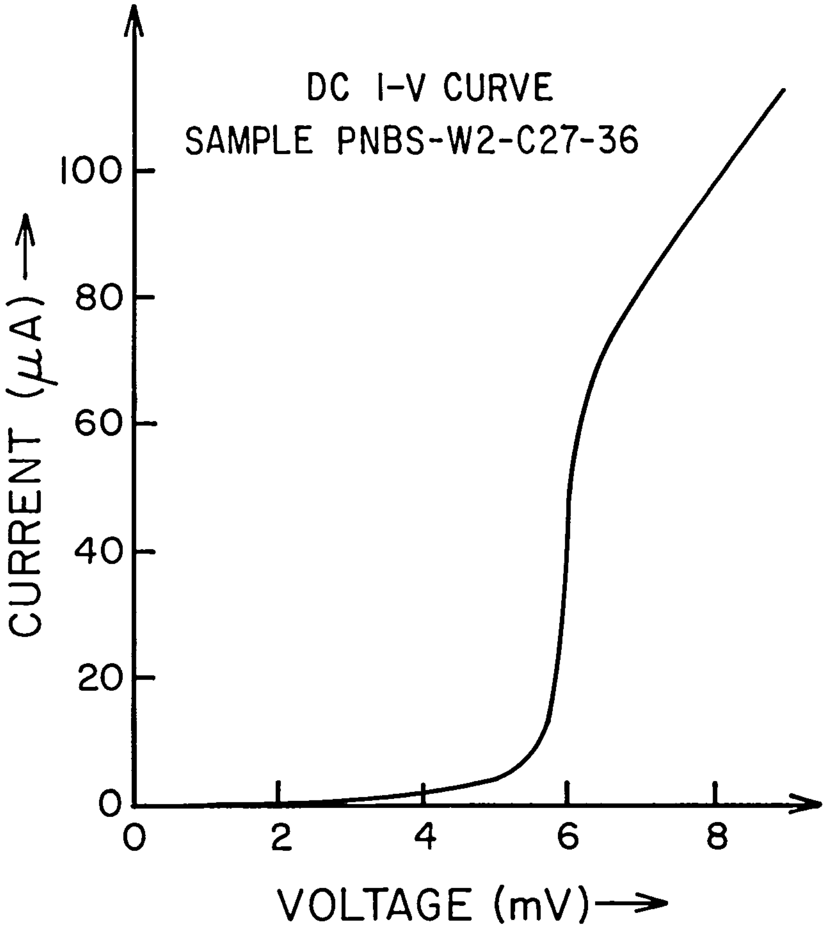

Fig. 9: DC I-V Curve Used in the Example in Section 4

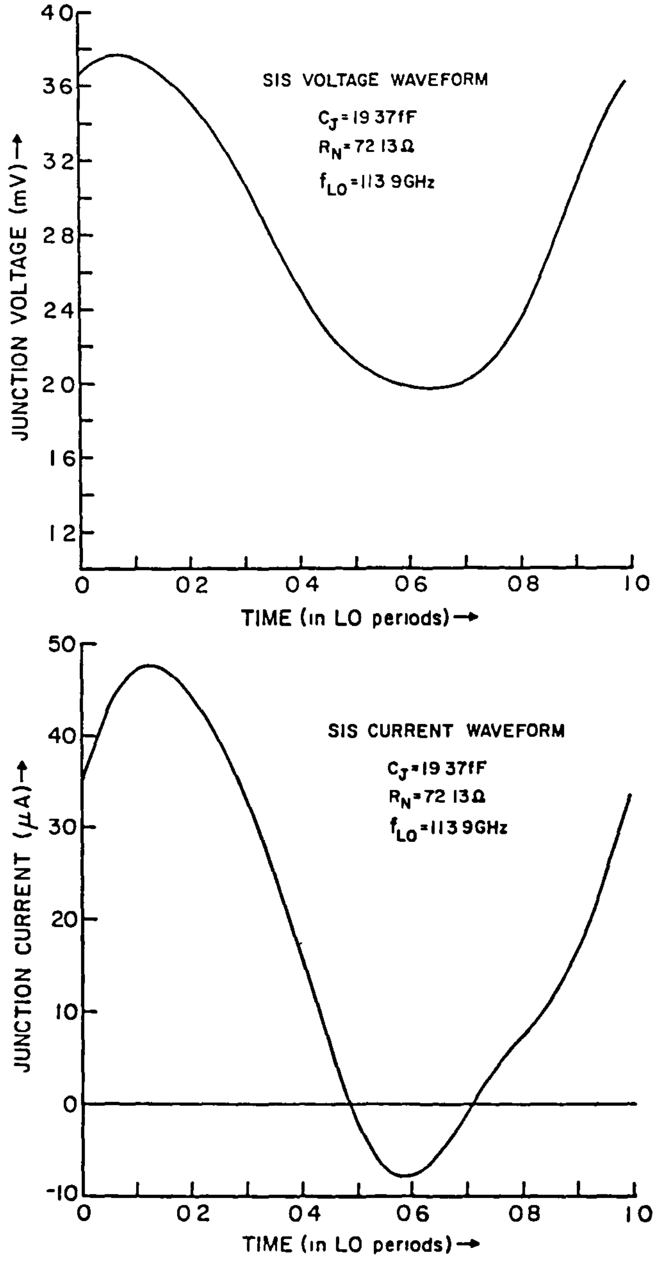

Fig. 10: Typical Junction Waveforms: (a) Voltage; (b) Current

5. Conclusions

A FORTRAN computer program has been developed for analyzing SIS mixers with arbitrary embedding impedances at all LO harmonics and sidebands. This program has been verified using the three-frequency approximation and the multiple reflection algorithm. Typical current and voltage waveforms at the SIS junction are plotted. Although a thorough study of the effect of varying junction capacitance has yet to be completed, in one specific example optimum mixer performance occurs for large or very small values of $\omega_p R_N C_J$. Low values of this parameter, roughly 0.3 to 2.0, lead to a deterioration in mixer performance.

Tests have also been performed on the relative speeds of the two nonlinear large signal analysis methods. For a typical SIS mixer situation, the voltage update method provides an order of magnitude improvement in total CPU time over that required by the multiple reflection technique.

APPENDIX A Supplementary Computer Program Information

A.1 Typical Run of Program SISJUN

The program SISJUN generates, without input, a typical single-junction DC I-V curve in digitized form. It is output on unit 7 and consists of 2049 lines of data. The first few lines of output are shown below. This data is to be used as input to GETCHI and is therefore given as a means for the verification of the correct functioning of programs GETCHI and SISCAP. Although SISJUN generates equally spaced data for input to program GETCHI, this is not a necessity for GETCHI input, as discussed in Section 3.2.

A.2 Typical Run of Program GETCHI

Given in this section is the output of a typical run of the program GETCHI. It should be remembered that three files have to be defined corresponding to FORTRAN unit 7 (input I-V data), unit 8 (output CHI data) and unit 9 (output I-V data). In the example shown here, the program uses as input on unit 7 the output of the program SISJUN.

The listing of the console output (given below) is self explanatory. It should be emphasized that, as a precaution, the two output files of CHI data and DC I-V data should be plotted to ensure there are no discontinuities in this data. Such a situation is possible when the input experimental data has been affected by noise during the measurement process. Such noise can destroy the monotonicity of the output DC I-V curve, a property relied upon by portions of the computer programs described in this report.

```
FILE: SISJUN   DOC      A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2


sisjun
FI Ø7 DISK SISJUN DAT A1 ( PERM RECFM FA LRECL 8Ø
EXEC FORTXCLG SISJUN XREF GOSTMT MAP

FORTRAN H EXTENDED (ENHANCED)

SOURCE STATEMENTS =       43, PROGRAM SIZE =      1476, SUBPROGRAM NAME =  MAIN
*** Files ' SISJUN TEXT ' and ' SISJUN LISTING ' have been created
*** Loading ... file ' LOAD MAP A ' is being created
EXECUTION BEGINS...
R; T=4.75/5.54 12:15:38
```

```
FILE: SISJUN   DATA     A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2


 2Ø48
 Ø.Ø               Ø.Ø
 Ø 49999999E-Ø5 Ø.16666675E-Ø8
 Ø.99999961E-Ø5 Ø.33333336E-Ø8
 Ø.14999998E-Ø4 Ø.5ØØØØØØ4E-Ø8
 Ø 19999992E-Ø4 Ø.66666672E-Ø8
 Ø.25ØØØØØ1E-Ø4 Ø.8333334ØE-Ø8
 Ø 29999996E-Ø4 Ø.1ØØØØØØ4E-Ø7
 Ø.3499999ØE-Ø4 Ø.11666668E-Ø7
 Ø.39999984E-Ø4 Ø.13333334E-Ø7
 Ø.44999979E-Ø4 Ø.14999998E-Ø7
 Ø.49999973E-Ø4 Ø.16666664E-Ø7
 Ø.54999997E-Ø4 Ø 18333336E-Ø7
 Ø.59999991E-Ø4 Ø.2ØØØØØØ6E-Ø7
 Ø.64999986E-Ø4 Ø.21666668E-Ø7
 Ø.6999998ØE-Ø4 Ø.23333335E-Ø7
 Ø.74999974E-Ø4 Ø 25ØØØØØ2E-Ø7
 Ø.79999998E-Ø4 Ø.26666668E-Ø7
 Ø.84999992E-Ø4 Ø.28333343E-Ø7
 Ø.89999987E-Ø4 Ø.29999995E-Ø7
 Ø.94999981E-Ø4 Ø 31666673E-Ø7
```

```
getchi
FI Ø7 DISK SISJUN DAT A1 ( PERM RECFM FA LRECL 8Ø
FI Ø8 DISK CHI DAT A1 ( PERM RECFM FA LRECL 8Ø
FI Ø9 DISK IV DAT A1 ( PERM RECFM FA LRECL 8Ø
EXEC FORTXCLG GETCHI XREF GOSTMT MAP

FORTRAN H EXTENDED (ENHANCED)

SOURCE STATEMENTS =      31, PROGRAM SIZE =    67Ø42, SUBPROGRAM NAME =  MAIN

SOURCE STATEMENTS =      63, PROGRAM SIZE =     2ØØ4, SUBPROGRAM NAME =  EVEN

SOURCE STATEMENTS =      1Ø, PROGRAM SIZE =      494, SUBPROGRAM NAME =REFINE

SOURCE STATEMENTS =      77, PROGRAM SIZE =     2542, SUBPROGRAM NAME = INEXP

SOURCE STATEMENTS =      56, PROGRAM SIZE =     1578, SUBPROGRAM NAME =IVTOCH

SOURCE STATEMENTS =      71, PROGRAM SIZE =     221Ø, SUBPROGRAM NAME =   FFT

SOURCE STATEMENTS =      15, PROGRAM SIZE =      4Ø4, SUBPROGRAM NAME = JUXTA

SOURCE STATEMENTS =      15, PROGRAM SIZE =      484, SUBPROGRAM NAME = FACNO

SOURCE STATEMENTS =      63, PROGRAM SIZE =     1978, SUBPROGRAM NAME = REEXP
*** Files ' GETCHI TEXT ' and ' GETCHI LISTING ' have been created
*** Loading ... file ' LOAD MAP A ' is being created
EXECUTION BEGINS...

THIS PROGRAM, GETCHI, PERFORMS A PREPROCESSING
FUNCTION ON THE INPUT DC I-V CURVE DATA PRIOR TO USING
THE PROGRAM SISCAP.  OUTPUT FROM GETCHI IS THE PRE-
PROCESSED DC I-V DATA, CHI IN AN APPROPRIATELY DIS-
CRETIZED FORM AND VALUES FOR THE PARAMETERS: NOCHI,
NOIV, NOPNTS AND RN

INPUT THE GAP VOLTAGE (VGAP) IN VOLTS OF THE SINGLE
       JUNCTION EQUIVALENT DEVICE (FORMAT E15.8)
Ø.ØØ3

INPUTTING THE INPUT I-V CURVE DATA FROM UNIT 7

INPUT THE NUMBER OF JUNCTIONS (NOJUN) IN DEVICE (I5)
    1

INPUT THE LO FREQUENCY IN HERTZ (FREQ) (E15.8)
       113.9e9

NUMBER OF LO CYCLES OF CHI IN SISCAP SHOULD BE  >     2
THE NUMBER OF POINTS PER LO CYCLE IN SISCAP SHOULD BE >    99

INPUT NUMBER OF LO CYCLES (NOCYC) (I5)
INPUT NO OF POINTS PER LO CYCLE (NOPNTS) (I5)
    4
  128


FILE: GETCHI   DOC      A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2


NOCHI =   512

INPUT DELTAV (TYP  Ø.2ØØØE-Ø4 VOLTS) (E15.8)
INPUT NUMBER OF POINTS REQUIRED (NOIV) (PWR OF 2) (TYP 1Ø24) (I5)
Ø.ØØØØ2
 1Ø24


INFORMATION FOR TRANSFERRAL TO PROGRAM SISCAP:
OUTPUTTING THE CHI FILE ON UNIT 8: NOCHI =   512
OUTPUTTING THE IV FILE ON UNIT 9:  NOIV  =  1Ø24
NOPNTS =   128
RN =  Ø.72131729E+Ø2 OHMS
R; T=13.93/15.93 12:17:46
```

```
FILE: CHI        DATA      A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2


  512
 Ø.Ø              Ø.Ø
 Ø.68591129E-13-Ø.23Ø19856E+Ø8
 Ø.13718226E-12-Ø.41334912E+Ø8
 Ø.2Ø577339E-12-Ø.52214416E+Ø8
 Ø.27436452E-12-Ø.55727536E+Ø8
 Ø.34295564E-12-Ø.54Ø98384E+Ø8
 Ø.41154677E-12-Ø.5Ø1163Ø4E+Ø8
 Ø.48Ø1379ØE-12-Ø.4565344ØE+Ø8
 Ø.548729Ø3E-12-Ø.411484ØØE+Ø8
 Ø.61732Ø16E-12-Ø.36Ø8Ø416E+Ø8
 Ø.68591129E-12-Ø.29899776E+Ø8
 Ø.7545Ø242E-12-Ø.22654ØØØE+Ø8
 Ø.823Ø9355E-12-Ø.14985496E+Ø8
 Ø.89168468E-12-Ø.7689Ø44ØE+Ø7
 Ø.96Ø27526E-12-Ø.128Ø337ØE+Ø7
 Ø.1Ø288662E-11 Ø.4Ø98Ø89ØE+Ø7
 Ø.1Ø974581E-11 Ø.8467352ØE+Ø7
 Ø.1166Ø49ØE-11 Ø.1176916ØE+Ø8
 Ø.123464ØØE-11 Ø.13812Ø84E+Ø8
 Ø 13Ø3231ØE-11 Ø.14426568E+Ø8
```

```
FILE: IV         DATA      A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2


 1Ø24
 Ø.Ø              Ø.Ø
 Ø.2ØØØØØØ7E-Ø4-Ø.27Ø6Ø435E-Ø6
 Ø.4ØØØØØ14E-Ø4-Ø.5412Ø761E-Ø6
 Ø.6ØØØØØ2ØE-Ø4-Ø.81181Ø88E-Ø6
 Ø.8ØØØØØ27E-Ø4-Ø.1Ø824133E-Ø5
 Ø.1ØØØØØØ3E-Ø3-Ø.1353Ø171E-Ø5
 Ø.12ØØØØØ4E-Ø3-Ø.162362Ø8E-Ø5
 Ø.14ØØØØØ5E-Ø3-Ø.18942246E-Ø5
 Ø.16ØØØØØ5E-Ø3-Ø.21648284E-Ø5
 Ø.18ØØØØØ6E-Ø3-Ø.24354322E-Ø5
 Ø.2ØØØØØØ7E-Ø3-Ø.27Ø6Ø36ØE-Ø5
 Ø.22ØØØØØ7E-Ø3-Ø.29766397E-Ø5
 Ø.24ØØØØØ8E-Ø3-Ø.32472444E-Ø5
 Ø.25999988E-Ø3-Ø.35178446E-Ø5
 Ø.2799998ØE-Ø3-Ø.37884465E-Ø5
 Ø.29999972E-Ø3-Ø.4Ø59Ø494E-Ø5
 Ø.31999964E-Ø3-Ø.43296513E-Ø5
 Ø.33999956E-Ø3-Ø.46ØØ2542E-Ø5
 Ø.35999948E-Ø3-Ø.487Ø8571E-Ø5
 Ø.3799994ØE-Ø3-Ø.514146ØØE-Ø5
```

```
FILE: SISCAP   DOC      A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

siscap
FI 11 DISK PARAM DAT A1 ( PERM RECFM FA LRECL 8Ø
FI Ø4 DISK CHI DAT A1 ( PERM RECFM FA LRECL 8Ø
FI Ø8 DISK IV DAT A1 ( PERM RECFM FA LRECL 8Ø
FI 14 DISK ZSSB DAT A1 ( PERM RECFM FA LRECL 8Ø
FI 15 DISK ZLO DAT A1 ( PERM RECFM FA LRECL 8Ø
EXEC FORTXCLG SISCAP GOSTMT XREF MAP

FORTRAN H EXTENDED (ENHANCED)

SOURCE STATEMENTS =      17Ø, PROGRAM SIZE =     42222, SUBPROGRAM NAME =   MAIN

SOURCE STATEMENTS =       47, PROGRAM SIZE =      181Ø, SUBPROGRAM NAME =GETSSB

SOURCE STATEMENTS =       39, PROGRAM SIZE =      1556, SUBPROGRAM NAME =   GETZ

SOURCE STATEMENTS =       25, PROGRAM SIZE =      2478, SUBPROGRAM NAME =LINEAR

SOURCE STATEMENTS =       68, PROGRAM SIZE =     2Ø228, SUBPROGRAM NAME = SISNL

SOURCE STATEMENTS =       27, PROGRAM SIZE =       734, SUBPROGRAM NAME =COMPAR

SOURCE STATEMENTS =       57, PROGRAM SIZE =     35Ø34, SUBPROGRAM NAME =   TOPS

SOURCE STATEMENTS =       26, PROGRAM SIZE =      1Ø54, SUBPROGRAM NAME =RESNOI

SOURCE STATEMENTS =       36, PROGRAM SIZE =      1754, SUBPROGRAM NAME =PROCES

SOURCE STATEMENTS =       79, PROGRAM SIZE =      4476, SUBPROGRAM NAME = SIGNL

SOURCE STATEMENTS =       31, PROGRAM SIZE =     176Ø8, SUBPROGRAM NAME =   GETJ

SOURCE STATEMENTS =       16, PROGRAM SIZE =       762, SUBPROGRAM NAME =SIGLIN

SOURCE STATFMENTS =       45, PROGRAM SIZE =      243Ø, SUBPROGRAM NAME =LOSCAL

SOURCE STATEMENTS =       28, PROGRAM SIZE =       866, SUBPROGRAM NAME = EVALJ

SOURCE STATEMENTS =       4Ø, PROGRAM SIZE =      1818, SUBPROGRAM NAME =SISNOI

SOURCE STATEMENTS =       23, PROGRAM SIZE =       998, SUBPROGRAM NAME = INVRT

SOURCE STATEMENTS =       25, PROGRAM SIZE =      1238, SUBPROGRAM NAME =FACTOR

SOURCE STATEMENTS =       25, PROGRAM SIZE =      1418, SUBPROGRAM NAME = SUBST

SOURCE STATEMENTS =       71, PROGRAM SIZE =      221Ø, SUBPROGRAM NAME =    FFT

SOURCE STATEMENTS =       15, PROGRAM SIZE =       4Ø4, SUBPROGRAM NAME = JUXTA

SOURCE STATEMENTS =       15, PROGRAM SIZE =       484, SUBPROGRAM NAME = FACNO

SOURCE STATEMENTS =       15, PROGRAM SIZE =       576, SUBPROGRAM NAME =   ZERO

SOURCE STATEMENTS =      14Ø, PROGRAM SIZE =      4574, SUBPROGRAM NAME =THEORY
```

```
FILE: SISCAP   DOC      A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2


SOURCE STATEMENTS =       26, PROGRAM SIZE =       962, SUBPROGRAM NAME = DERIJ

SOURCE STATEMENTS =       66, PROGRAM SIZE =      16ØØ, SUBPROGRAM NAME =  BESJ

SOURCE STATEMENTS =       12, PROGRAM SIZE =       396, SUBPROGRAM NAME =  COTH

SOURCE STATEMENTS =       56, PROGRAM SIZE =      3Ø42, SUBPROGRAM NAME =GEN3PT

SOURCE STATEMENTS =       65, PROGRAM SIZE =      1754, SUBPROGRAM NAME = TERP5

SOURCE STATEMENTS =       65, PROGRAM SIZE =      6928, SUBPROGRAM NAME =   MRT

SOURCE STATEMENTS =       2Ø, PROGRAM SIZE =     12252, SUBPROGRAM NAME =SISCHA

SOURCE STATEMENTS =      11Ø, PROGRAM SIZE =      445Ø, SUBPROGRAM NAME =   SIS

SOURCE STATEMENTS =       12, PROGRAM SIZE =       43Ø, SUBPROGRAM NAME =YFUNCT

SOURCE STATEMENTS =       38, PROGRAM SIZE =       854, SUBPROGRAM NAME = COMP2

SOURCE STATEMENTS =        8, PROGRAM SIZE =       388, SUBPROGRAM NAME =FIXPHA
*** Files ' SISCAP TEXT ' and ' SISCAP LISTING ' have been created
*** Loading ... file ' LOAD MAP A ' is being created
EXECUTION BEGINS...

FREQ =  Ø.1139ØØØØE+12HZ FIF =  Ø.1ØØØØØØØE+Ø7HZ
CAP  =  Ø.19371999E-13F  RN  =  Ø.72131729E+Ø2OHMS
TEMP =  Ø.26999998E+Ø1K
IBASE2 =     1Ø NOIV   =  1Ø24
IBASE1 =      7 NOPNTS =   128
MMAX   =     31 NMAX   =    31
ISIG   =      2 NOPER  =    5Ø
NOCHI  =    512 NODAT  =     1
TOL =  Ø.1ØØØØØØØE-Ø3           EPS =  Ø.1ØØØØØØØE-Ø3
DELV =  Ø.25ØØØØØ1E-Ø4VOLTS     VSTART =  Ø.27999999E-Ø2VOLTS
VLOSTR =  Ø.35ØØØØØ1E-Ø2VOLTS DELVLO =  Ø.1ØØØØØØØE-Ø3VOLTS
NLO =     1                     P =  Ø.1ØØØØØØØE+Ø1
RID =  Ø.5ØØØØØØØE+Ø2OHMS       ZØ =  Ø.5ØØØØØØØE+Ø2OHMS

INPUTTING THE CHI FUNCTION
DELTAT =  Ø.68597905E-13 SECS
NUMBER OF POINTS READ =   512

INPUTTING THE SIS IV CHARACTERISTIC
DELTAV =  Ø.19997358E-Ø4 VOLTS
NUMBER OF POINTS READ =  1Ø24

CALCULATING THE VECTOR J

INPUTTING THE LO HARMONIC IMPEDANCES
THE FIRST  1Ø LO HARM IMPEDANCES (EXC CAPAC) ARE:
    Ø  Ø.Ø            +J Ø.Ø            OHMS
    1  Ø.55ØØØØØØE+Ø2+J Ø.92ØØØØØØE+Ø2OHMS
    2  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
    3  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
    4  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
    5  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
    6  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
    7  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
    8  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
    9  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
Y PUMP (INCL CAPAC) =  Ø.47871843E-Ø2 +J -Ø.8ØØ766Ø5E-Ø2S
NUMBER OF POINTS READ =    64

INPUTTING SIDEBAND TERMINATION INFORMATION
THE FIRST  1Ø SIDEBAND IMPEDANCES (EXC CAPAC):
    Ø  Ø.5ØØØØØØØE+Ø2+J Ø.Ø           OHMS
    1  Ø.55ØØØØØØE+Ø2+J Ø.92ØØØØØØE+Ø2OHMS
   -1  Ø.55ØØØØØØE+Ø2+J Ø.92ØØØØØØE+Ø2OHMS
    2  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
   -2  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
    3  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
   -3  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
    4  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
   -4  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
    5  Ø.7213ØØØ5E+Ø2+J Ø.99999994E-1ØOHMS
Y SIGNAL (INCL CAPAC) =  Ø.47871843E-Ø2 +J -Ø.8ØØ766Ø5E-Ø2S
Y IMAGE  (INCL CAPAC) =  Ø.47871843E-Ø2 +J  Ø.8ØØ766Ø5E-Ø2S
NUMBER OF POINTS READ =   127

**********************************
**********CASE    1***************
**********************************


EMBEDDING THEVENIN VDC SOURCE        =  Ø.27999997E-Ø2 VOLTS
EMBEDDING THEVENIN VLO SOURCE (PEAK) =  Ø.34999999E-Ø2 VOLTS

INCIDENT LOCAL OSCILLATOR POWER =  Ø.2784Ø887E-Ø7 WATTS


GENERAL NONLINEAR ANALYSIS USING VOLTAGE UPDATE METHOD
BEGINNING THE ITERATION PHASE

       ITERATION NUMBER =      1
       ERR =  Ø.1ØØØØØØØE+Ø1POSN NO =     1

       ITERATION NUMBER =      2
       ERR =  Ø.139Ø8285E+ØØPOSN NO =    74

       ITERATION NUMBER =      3
       ERR =  Ø.7846272ØE-Ø1POSN NO =    85

       ITERATION NUMBER =      4
       ERR =  Ø.5975328ØE-Ø1POSN NO =    92

       ITERATION NUMBER =      5
       ERR =  Ø.1662169ØE-Ø1POSN NO =    99
```

```
FILE: SISCAP   DOC      A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

         ITERATION NUMBER =      6
         ERR =  Ø.538646Ø6E-Ø2POSN NO =     85

         ITERATION NUMBER =      7
         ERR =  Ø.31ØØ89Ø9E-Ø2POSN NO =     93

         ITERATION NUMBER =      8
         ERR =  Ø.983Ø7454E-Ø3POSN NO =     97

         ITERATION NUMBER =      9
         ERR =  Ø.3879671ØE-Ø3POSN NO =     93

         ITERATION NUMBER =     1Ø
         ERR =  Ø.18293827E-Ø3POSN NO =     94

         ITERATION NUMBER =     11
         ERR =  Ø.61962215E-Ø4POSN NO =     95

LARGE SIGNAL ANALYSIS COMPLETED IN    11 ITNS

POINT NO. =      1VOLTAGE =  Ø.36563654E-Ø2CURRENT =  Ø.351Ø2436E-Ø4
POINT NO. =      2VOLTAGE =  Ø.36812625E-Ø2CURRENT =  Ø.36557831E-Ø4
POINT NO. =      3VOLTAGE =  Ø.37Ø26438E-Ø2CURRENT =  Ø.379542Ø3E-Ø4
POINT NO. =      4VOLTAGE =  Ø.372Ø5843E-Ø2CURRENT =  Ø.39285Ø34E-Ø4
POINT NO. =      5VOLTAGE =  Ø.37351914E-Ø2CURRENT =  Ø.4Ø5322Ø6E-Ø4
POINT NO. =      6VOLTAGE =  Ø.37465971E-Ø2CURRENT =  Ø.41695879E-Ø4
POINT NO. =      7VOLTAGE =  Ø.3754958ØE-Ø2CURRENT =  Ø.42753862E-Ø4
POINT NO. =      8VOLTAGE =  Ø.376Ø4398E-Ø2CURRENT =  Ø.43715656E-Ø4
POINT NO. =      9VOLTAGE =  Ø.37632226E-Ø2CURRENT =  Ø.44562665E-Ø4
POINT NO. =     1ØVOLTAGE =  Ø.37634813E-Ø2CURRENT =  Ø.453Ø4492E-Ø4
POINT NO. =     11VOLTAGE =  Ø.37613946E-Ø2CURRENT =  Ø.45932538E-Ø4
POINT NO. =     12VOLTAGE =  Ø.375713Ø8E-Ø2CURRENT =  Ø.46451751E-Ø4
POINT NO. =     13VOLTAGE =  Ø.375Ø8472E-Ø2CURRENT =  Ø.46863ØØ2E-Ø4
POINT NO. =     14VOLTAGE =  Ø.37426832E-Ø2CURRENT =  Ø.47166192E-Ø4
POINT NO. =     15VOLTAGE =  Ø.37327681E-Ø2CURRENT =  Ø.4737Ø1Ø7E-Ø4
POINT NO. =     16VOLTAGE =  Ø.37212Ø6ØE-Ø2CURRENT =  Ø.474755Ø7E-Ø4
POINT NO. =     17VOLTAGE =  Ø.37Ø8Ø892E-Ø2CURRENT =  Ø.47486261E-Ø4
POINT NO. =     18VOLTAGE =  Ø.36934826E-Ø2CURRENT =  Ø.47415Ø29E-Ø4
POINT NO. =     19VOLTAGE =  Ø.3677445ØE-Ø2CURRENT =  Ø.47258392E-Ø4
POINT NO. =     2ØVOLTAGE =  Ø.366ØØ123E-Ø2CURRENT =  Ø.47Ø261ØØE-Ø4
POINT NO. =     21VOLTAGE =  Ø.364121ØØE-Ø2CURRENT =  Ø.46722Ø96E-Ø4
POINT NO. =     22VOLTAGE =  Ø.3621Ø5Ø4E-Ø2CURRENT =  Ø.46355111E-Ø4
POINT NO. =     23VOLTAGE =  Ø.359954Ø6E-Ø2CURRENT =  Ø.45924549E-Ø4
POINT NO. =     24VOLTAGE =  Ø.35766761E-Ø2CURRENT =  Ø.45438268E-Ø4
POINT NO. =     25VOLTAGE =  Ø.35524534E-Ø2CURRENT =  Ø.44896689E-Ø4
POINT NO. =     26VOLTAGE =  Ø.3526862ØE-Ø2CURRENT =  Ø.443Ø8676E-Ø4
POINT NO. =     27VOLTAGE =  Ø.34998977E-Ø2CURRENT =  Ø.4367Ø356E-Ø4
POINT NO. =     28VOLTAGE =  Ø.34715536E-Ø2CURRENT =  Ø.42993Ø66E-Ø4
POINT NO. =     29VOLTAGE =  Ø.34418299E-Ø2CURRENT =  Ø.42271175E-Ø4
POINT NO. =     3ØVOLTAGE =  Ø.341Ø7291E-Ø2CURRENT =  Ø.415Ø9833E-Ø4
POINT NO. =     31VOLTAGE =  Ø.33782658E-Ø2CURRENT =  Ø.4Ø7Ø9172E-Ø4
POINT NO. =     32VOLTAGE =  Ø.33444576E-Ø2CURRENT =  Ø.39871244E-Ø4
POINT NO. =     33VOLTAGE =  Ø.33Ø93365E-Ø2CURRENT =  Ø.389969Ø6E-Ø4
POINT NO. =     34VOLTAGE =  Ø.3272936ØE-Ø2CURRENT =  Ø.38Ø839Ø5E-Ø4
POINT NO. =     35VOLTAGE =  Ø.323531Ø5E-Ø2CURRENT =  Ø.37137594E-Ø4
```

```
FILE: SISCAP   DOC      A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

POINT NO. =    36VOLTAGE =  Ø.319652Ø7E-Ø2CURRENT =  Ø.36154655E-Ø4
POINT NO. =    37VOLTAGE =  Ø.315664Ø1E-Ø2CURRENT =  Ø.35135294E-Ø4
POINT NO. =    38VOLTAGE =  Ø.31157513E-Ø2CURRENT =  Ø.34Ø81284E-Ø4
POINT NO. =    39VOLTAGE =  Ø.3Ø739547E-Ø2CURRENT =  Ø.32992626E-Ø4
POINT NO. =    4ØVOLTAGE =  Ø.3Ø313574E-Ø2CURRENT =  Ø.31867166E-Ø4
POINT NO. =    41VOLTAGE =  Ø.298ØØ819E-Ø2CURRENT =  Ø.3Ø7Ø55Ø2E-Ø4
POINT NO. =    42VOLTAGE =  Ø.29442557E-Ø2CURRENT =  Ø.295Ø8359E-Ø4
POINT NO. =    43VOLTAGE =  Ø.29ØØØ2Ø5E-Ø2CURRENT =  Ø.28276278E-Ø4
POINT NO. =    44VOLTAGE =  Ø.28555247E-Ø2CURRENT =  Ø.27ØØ9199E-Ø4
POINT NO. =    45VOLTAGE =  Ø.281Ø9262E-Ø2CURRENT =  Ø.257Ø67ØØE-Ø4
POINT NO. =    46VOLTAGE =  Ø.27663852E-Ø2CURRENT =  Ø.24371766E-Ø4
POINT NO. =    47VOLTAGE =  Ø.2722Ø668E-Ø2CURRENT =  Ø.23ØØ24Ø2E-Ø4
POINT NO. =    48VOLTAGE =  Ø.26781391E-Ø2CURRENT =  Ø.216Ø6917E-Ø4
POINT NO. =    49VOLTAGE =  Ø.26347695E-Ø2CURRENT =  Ø.2Ø178Ø2ØE-Ø4
POINT NO. =    5ØVOLTAGE =  Ø.25921215E-Ø2CURRENT =  Ø.1873Ø527E-Ø4
POINT NO. =    51VOLTAGE =  Ø.255Ø3556E-Ø2CURRENT =  Ø.17258368E-Ø4
POINT NO. =    52VOLTAGE =  Ø.25Ø96249E-Ø2CURRENT =  Ø.15773418E-Ø4
POINT NO. =    53VOLTAGE =  Ø.247ØØ745E-Ø2CURRENT =  Ø.14274468E-Ø4
POINT NO. =    54VOLTAGE =  Ø.2431835ØE-Ø2CURRENT =  Ø.12772449E-Ø4
POINT NO. =    55VOLTAGE =  Ø.2395Ø271E-Ø2CURRENT =  Ø.112692Ø7E-Ø4
POINT NO. =    56VOLTAGE =  Ø.23597558E-Ø2CURRENT =  Ø.9775Ø744E-Ø5
POINT NO. =    57VOLTAGE =  Ø.23261Ø76E-Ø2CURRENT =  Ø.8293ØØ55E-Ø5
POINT NO. =    58VOLTAGE =  Ø.22941544E-Ø2CURRENT =  Ø.68375812E-Ø5
POINT NO. =    59VOLTAGE =  Ø.22639451E-Ø2CURRENT =  Ø.54113771E-Ø5
POINT NO. =    6ØVOLTAGE =  Ø.22355143E-Ø2CURRENT =  Ø.4Ø259911E-Ø5
POINT NO. =    61VOLTAGE =  Ø.22Ø88757E-Ø2CURRENT =  Ø.26851Ø48E-Ø5
POINT NO. =    62VOLTAGE =  Ø.2184Ø241E-Ø2CURRENT =  Ø.14Ø31393E-Ø5
POINT NO. =    63VOLTAGE =  Ø.216Ø9364E-Ø2CURRENT =  Ø.18249557E-Ø6
POINT NO. =    64VOLTAGE =  Ø.21395762E-Ø2CURRENT = -Ø.966Ø4344E-Ø6
POINT NO. =    65VOLTAGE =  Ø.21198881E-Ø2CURRENT = -Ø.2Ø4271Ø5E-Ø5
POINT NO. =    66VOLTAGE =  Ø.21Ø18125E-Ø2CURRENT = -Ø.3Ø3Ø9Ø2ØE-Ø5
POINT NO. =    67VOLTAGE =  Ø.2Ø852718E-Ø2CURRENT = -Ø.39313Ø18E-Ø5
POINT NO. =    68VOLTAGE =  Ø.2Ø7Ø1862E-Ø2CURRENT = -Ø.47391368E-Ø5
POINT NO. =    69VOLTAGE =  Ø.2Ø564718E-Ø2CURRENT = -Ø.5453Ø1Ø1E-Ø5
POINT NO. =    7ØVOLTAGE =  Ø.2Ø44Ø421E-Ø2CURRENT = -Ø.6Ø664897E-Ø5
POINT NO. =    71VOLTAGE =  Ø.2Ø328118E-Ø2CURRENT = -Ø.65798959E-Ø5
POINT NO. =    72VOLTAGE =  Ø.2Ø227Ø18E-Ø2CURRENT = -Ø.69922826E-Ø5
POINT NO. =    73VOLTAGE =  Ø.2Ø136358E-Ø2CURRENT = -Ø.73Ø88158E-Ø5
POINT NO. =    74VOLTAGE =  Ø.2ØØ555Ø3E-Ø2CURRENT = -Ø.752171Ø4E-Ø5
POINT NO. =    75VOLTAGE =  Ø.19983889E-Ø2CURRENT = -Ø.76428259E-Ø5
POINT NO. =    76VOLTAGE =  Ø.19921Ø69E-Ø2CURRENT = -Ø.76624419E-Ø5
POINT NO. =    77VOLTAGE =  Ø.1986674ØE-Ø2CURRENT = -Ø.759933Ø3E-Ø5
POINT NO. =    78VOLTAGE =  Ø.1982Ø742E-Ø2CURRENT = -Ø.74446871E-Ø5
POINT NO. =    79VOLTAGE =  Ø.19783Ø31E-Ø2CURRENT = -Ø.72172115E-Ø5
POINT NO. =    8ØVOLTAGE =  Ø.19753748E-Ø2CURRENT = -Ø.69Ø29Ø47E-Ø5
POINT NO. =    81VOLTAGE =  Ø.19733123E-Ø2CURRENT = -Ø.65317727E-Ø5
POINT NO. =    82VOLTAGE =  Ø.19721638E-Ø2CURRENT = -Ø.6Ø873426E-Ø5
POINT NO. =    83VOLTAGE =  Ø.19719815E-Ø2CURRENT = -Ø.5593276ØE-Ø5
POINT NO. =    84VOLTAGE =  Ø.19728362E-Ø2CURRENT = -Ø.5Ø428Ø62E-Ø5
POINT NO. =    85VOLTAGE =  Ø.19748122E-Ø2CURRENT = -Ø.44538319E-Ø5
POINT NO. =    86VOLTAGE =  Ø.1978ØØ74E-Ø2CURRENT = -Ø.3825844ØE-Ø5
POINT NO. =    87VOLTAGE =  Ø.19825266E-Ø2CURRENT = -Ø.3172Ø265E-Ø5
POINT NO. =    88VOLTAGE =  Ø.19884899E-Ø2CURRENT = -Ø.24886249E-Ø5
POINT NO. =    89VOLTAGE =  Ø.1996Ø22ØE-Ø2CURRENT = -Ø.18Ø43793E-Ø5
POINT NO. =    9ØVOLTAGE =  Ø.2ØØ52569E-Ø2CURRENT = -Ø.1Ø99Ø625E-Ø5
POINT NO. =    91VOLTAGE =  Ø.2Ø163292E-Ø2CURRENT = -Ø.394677Ø4E-Ø6
POINT NO. =    92VOLTAGE =  Ø.2Ø293787E-Ø2CURRENT =  Ø.314918ØØE-Ø6
POINT NO. =    93VOLTAGE =  Ø.2Ø445397E-Ø2CURRENT =  Ø.1Ø1Ø5286E-Ø5
POINT NO. =    94VOLTAGE =  Ø.2Ø619468E-Ø2CURRENT =  Ø.16979Ø29E-Ø5
POINT NO. =    95VOLTAGE =  Ø.2Ø817227E-Ø2CURRENT =  Ø.237Ø6962E-Ø5
POINT NO. =    96VOLTAGE =  Ø.21Ø39834E-Ø2CURRENT =  Ø.3Ø33987ØE-Ø5
POINT NO. =    97VOLTAGE =  Ø.2128826ØE-Ø2CURRENT =  Ø.36715355E-Ø5
POINT NO. =    98VOLTAGE =  Ø.215634Ø3E-Ø2CURRENT =  Ø.42979227E-Ø5
POINT NO. =    99VOLTAGE =  Ø.21865845E-Ø2CURRENT =  Ø.49Ø86666E-Ø5
POINT NO. =   1ØØVOLTAGE =  Ø.22196Ø25E-Ø2CURRENT =  Ø.54999255E-Ø5
POINT NO. =   1Ø1VOLTAGE =  Ø.22554Ø81E-Ø2CURRENT =  Ø.6Ø887687E-Ø5
POINT NO. =   1Ø2VOLTAGE =  Ø.22939951E-Ø2CURRENT =  Ø.66628563E-Ø5
POINT NO. =    1Ø3VOLTAGE =  Ø.23353205E-Ø2CURRENT =  Ø.72361436E-Ø5
POINT NO. =    1Ø4VOLTAGE =  Ø.23793189E-Ø2CURRENT =  Ø.78Ø79611E-Ø5
POINT NO. =    1Ø5VOLTAGE =  Ø.24258881E-Ø2CURRENT =  Ø.8387Ø545E-Ø5
POINT NO. =    1Ø6VOLTAGE =  Ø.24749Ø34E-Ø2CURRENT =  Ø.89761743E-Ø5
POINT NO. =    1Ø7VOLTAGE =  Ø.25262Ø27E-Ø2CURRENT =  Ø.95865689E-Ø5
POINT NO. =    1Ø8VOLTAGE =  Ø.25795978E-Ø2CURRENT =  Ø.1Ø218384E-Ø4
POINT NO. =    1Ø9VOLTAGE =  Ø.26348669E-Ø2CURRENT =  Ø.1Ø884571E-Ø4
POINT NO. =    11ØVOLTAGE =  Ø.26917665E-Ø2CURRENT =  Ø.11585973E-Ø4
POINT NO. =    111VOLTAGE =  Ø.275ØØ226E-Ø2CURRENT =  Ø.12333Ø83E-Ø4
POINT NO. =    112VOLTAGE =  Ø.26Ø93394E-Ø2CURRENT =  Ø.1312932ØE-Ø4
POINT NO. =    113VOLTAGE =  Ø.28693988E-Ø2CURRENT =  Ø.13978235E-Ø4
POINT NO. =    114VOLTAGE =  Ø.29298668E-Ø2CURRENT =  Ø.148887Ø5E-Ø4
POINT NO. =    115VOLTAGE =  Ø.299Ø3932E-Ø2CURRENT =  Ø.15862883E-Ø4
POINT NO. =    116VOLTAGE =  Ø.3Ø5Ø619ØE-Ø2CURRENT =  Ø.169Ø3679E-Ø4
POINT NO. =    117VOLTAGE =  Ø.311Ø178ØE-Ø2CURRENT =  Ø.18Ø1Ø862E-Ø4
POINT NO. =    118VOLTAGE =  Ø.31687Ø77E-Ø2CURRENT =  Ø.19187792E-Ø4
POINT NO. =    119VOLTAGE =  Ø.32258469E-Ø2CURRENT =  Ø.2Ø43Ø685E-Ø4
POINT NO. =    12ØVOLTAGE =  Ø.32812464E-Ø2CURRENT =  Ø.21738233E-Ø4
POINT NO. =    121VOLTAGE =  Ø.33345749E-Ø2CURRENT =  Ø.231Ø39Ø1E-Ø4
POINT NO. =    122VOLTAGE =  Ø.33855219E-Ø2CURRENT =  Ø.24523295E-Ø4
POINT NO. =    123VOLTAGE =  Ø.34338Ø73E-Ø2CURRENT =  Ø.25988586E-Ø4
POINT NO. =    124VOLTAGE =  Ø.347918Ø2E-Ø2CURRENT =  Ø.27487957E-Ø4
POINT NO. =    125VOLTAGE =  Ø.35214273E-Ø2CURRENT =  Ø.29Ø13652E-Ø4
POINT NO. =    126VOLTAGE =  Ø.356Ø3768E-Ø2CURRENT =  Ø.3Ø549549E-Ø4
POINT NO. =    127VOLTAGE =  Ø.35959ØØ5E-Ø2CURRENT =  Ø.32Ø87453E-Ø4
POINT NO. =    128VOLTAGE =  Ø.36279Ø93E-Ø2CURRENT =  Ø.3361Ø151E-Ø4
DC VOLTAGE AT THE JUNCTION         =  Ø.27999971E-Ø2 VOLTS
LO VOLTAGE (PEAK) AT THE JUNCTION =  Ø.92197349E-Ø3 VOLTS

SMALL SIGNAL AND NOISE ANALYSIS SECTION


DETERMINING SMALL SIGNAL MIXER PROPERTIES USING THE GENERAL ANALYSIS

       IF OUTPUT IMPEDANCE =  Ø.548180Ø91E+Ø3 +J -Ø.2379607Ø9E-Ø2
       IF VSWR REFERRED TO 5Ø.ØØ OHMS =  Ø.1Ø963613E+Ø2
       OUTPUT MISMATCH GAIN =  Ø.3Ø64ØØ12E+ØØ (-Ø.51371Ø98E+Ø1DB)

       INPUT SIGNAL IMPEDANCE =  Ø.8526265ØE+Ø2 +J  Ø.93624878E+Ø1OHMS
       INPUT MISMATCH GAIN =  Ø.82252783E+ØØ (-Ø.84849358E+ØØDB)
       INPUT GAIN CALCULATION REFERRED TO =  Ø.2Ø8891Ø5E+Ø3 OHMS

       CONVERSION GAIN (USB) EQUALS -Ø.3Ø229321E+Ø1 DB
       CONVERSION GAIN (LSB) EQUALS -Ø.3Ø229263E+Ø1 DB

       SSB TEMP OF MIXER (USB) =  Ø.23333755E+Ø2 DEG K
       SSB TEMP OF MIXER (LSB) =  Ø.2333371ØE+Ø2 DEG K
R; T=244.59/252.26 14:Ø3:14


FILE: PARAM    DAT      A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2


    2                         ISWIT
    1                         NODAT
       Ø.Ø25E-3               DELV
      2.8ØØØE-3               VSTART
     3.5ØØØØE-3               VLOSTR
     Ø.1ØØØØE-3               DELVLO
    1                         NLO
         1.ØE-4               TOL
         1.ØE-4               EPS
  128                         NOPNTS
    7                         IBASE1
 1Ø24                         NOIV
   1Ø                         IBASE2
  512                         NCCHI
   5Ø                         NCPER
   31                         MMAX
   31                         NMAX
      72.131729               RN
    19.372ØE-15               CAP
            2.7               TEMP
        113.9E9               FREQ
          1.ØE6               FIF
            1.Ø               P
         5Ø.ØØØ               RID
           5Ø.Ø               ZØ
```

```
FILE: ZSSB      DATA      A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

 127
            5Ø.Ø           Ø.Ø
       55.ØØØØØ      92.ØØØØØ
       55.ØØØØØ      92.ØØØØØ
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
```

```
FILE: ZLO       DATA      A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

  64
            Ø.Ø           Ø.Ø
          55.ØØ         92.ØØ
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
          72.13        1.E-1Ø
```

A.3 Typical Run of Program SISCAP

The following pages show a typical run of program SISCAP. It utilizes the CHI and DC I-V data produced from the run of GETCHI shown in the previous section. This program uses as input the following files: CHI data (unit no. 4), processed DC I-V data (unit no. 8), control parameters (unit no. 11), sideband termination impedances (unit no. 14) and LO harmonic impedances (unit no. 15). One output file is generated via the console.

A.4 Memory Allocation Requirements of GETCHI

Memory is allocated in program GETCHI in accordance with the following rules. NUM is preset by subroutine EVEN to be 4096. NOCHI and NOIV are numbers which the user obtains by running program GETCHI. Since the user needs the value of these parameters prior to running the program (in order to allocate the correct amount of memory for the program), the program may have to be run twice, the first time with an excessive amount of memory in order to obtain the values of NOCHI and NOIV. Table 2 below indicates the correct memory allocation required and in which subroutine the memory space has to be declared.

VARIABLE	SUBROUTINE	SIZE
VOLT2	MAIN	NUM
CUR2	MAIN	NUM
VOLT3	MAIN	NOCHI
CUR3	MAIN	NOCHI
VOLT4	MAIN	NOIV
CUR4	MAIN	NOIV

TABLE 2

VARIABLE	SUBROUTINE	SIZE
TIME	MAIN	NOCHI
CHI	MAIN	NOCHI
CCHI	MAIN	2*NOCHI
FACTOR	MAIN	2*NOCHI

TABLE 2

A.5 Memory Allocation Requirements of SISCAP

This section outlines the memory allocation requirements of program SISCAP. There are 5 key parameter values which must be specified prior to allocating the appropriate memory, namely NOPNTS, NOIV, NOCHI, MMAX and NMAX.

NOPNTS and NOCHI are determined by the discretization constraints on CHI. NOPNTS is the number of points per LO cycle and NOCHI is the number of CHI points input to SISCAP. Both are provided by using the program GETCHI. NOPNTS must be a power of 2 and NOCHI, in addition to being a power of 2, must be equal to or greater than NOPNTS.

NOIV is the number of points in the I-V curve input to the program SISCAP. It must also be a power of 2 and must be large enough such that the discretization of the I-V curve is adequate. This value is obtained from program GETCHI.

MMAX indicates the number of LO sidebands which are taken into account. MMAX must be odd and is limited to NOPNTS/2 - 1 in value. See Section 3.4 item (j) for a more detailed discussion.

NMAX (assumed odd) indicates how many terms are significant in the series which are used to calculate the small signal conversion properties of the mixer. Its value should be as discussed in item (k) of Section 3.4.

Having established these values, the memory allocation is as per Table 4. Note that the symbol '*' indicates a multiplication whereas the symbol 'x' indicates a multidimensional array.

VARIABLE	SUBROUTINE	SIZE
ZIMPED	MAIN	NOPNTS/2
VSOURC	MAIN	NOPNTS/2
CVAL	MAIN	NOPNTS
VOLT	MAIN	NOPNTS
CUR	MAIN	NOPNTS
OLDVOL	MAIN	NOPNTS
CTEMP	LINEAR	NOPNTS
W	TOPS	NOPNTS
CVOL	SIGNL	NOPNTS
CVOL	SISNL	NOPNTS
FACTOR	MAIN	2*MAX OF (NOIV & NOCHI)
J	MAIN	NOIV
IVDAT	MAIN	NOIV
CCHI	SISNL	NOCHI*2
CTEMP	GETJ	NOIV*2

TABLE 4

VARIABLE	SUBROUTINE	SIZE
ZSSB	MAIN	NOPNTS-1
CHI	MAIN	NOCHI
UTERM	SISNL	NOCHI*2
YSIG	TOPS	MMAX x MMAX
NOISE	TOPS	MMAX x MMAX
ZSIG	TOPS	MMAX x MMAX
B	TOPS	MMAX
STORE	TOPS	MMAX x MMAX
STORE	PROCES	MMAX
YSI2	TOPS	3 x 3
NOI2	TOPS	3 x 3
ZSI2	TOPS	3 x 3
VAL	THEORY	NMAX+10
STORE	TERP5	4
ESOURC	MRT	NOPNTS
ELEFT	MRT	NOPNTS
ERIGHT	MRT	NOPNTS
RHO	MRT	NOPNTS
CCUR	SISCHA	NOPNTS
ZTERM	SISCHA	NOPNTS
PHASE	SISCHA	NOCHI
VOLT2	SISCHA	NOCHI
UTERM	SISCHA	NOCHI
CVOL	SISNL	NOPNTS

TABLE 4 CONT.

A.6 Listing of Program SISJUN

This section gives the listing of the test data program SISJUN.

```
FILE: SISJUN   FORTRAN  A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

C
C     THIS PROGRAM PROVIDES A STANDARD TEST DC I-V CURVE
C     IN WHICH TO VERIFY THE CORRECT FUNCTIONING OF THE
C     PROGRAMS GETCHI AND SISCAP
C
C     I-V CURVE FOR FUNCTION PNBS W2 C2736 D4
C
      V1 = Ø.Ø
      V2 = 2.44
      V3 = 5.99
      V4 = 9.2Ø
C
      NOPNTS = 2Ø48
      DELTAV = Ø.Ø1
C
      WRITE(7,19) NOPNTS
19    FORMAT(I5)
C
      VSPAN = FLOAT(NOPNTS-1) * DELTAV
C
      I1 = 1
      I2M1 = V2 / DELTAV + Ø.5
      I2 = I2M1 + 1
      I3M1 = V3 / DELTAV + Ø.5
      I3 = I3M1 + 1
      I4M1 = V4 / DELTAV + Ø.5
      I4 = I4M1 + 1
C
      DO 1ØØ I = I1,I2M1
      VOUT = (FLOAT(I-1) * DELTAV)
      CUROUT = (VOUT/3.Ø) * 1.ØE-6 / 2.Ø
      VOUT = VOUT * 1.ØE-3 / 2.Ø
      WRITE(7,11Ø) VOUT, CUROUT
11Ø   FORMAT(2E15.8)
1ØØ   CONTINUE
C
      DO 2ØØ I = I2,I3M1
      VOUT = (FLOAT(I-1) * DELTAV)
      CUROUT = (Ø.9*VOUT - 1.37 + EXP((VOUT-4.98)*3.Ø775) +
     1 EXP((VOUT-5.749)*12.61)) * 1.ØE-6 / 2.Ø
      VOUT = VOUT * 1.ØE-3 / 2.Ø
      WRITE(7,11Ø) VOUT, CUROUT
2ØØ   CONTINUE
C
      DO 3ØØ I = I3,I4M1
      VOUT = (FLOAT(I-1) * DELTAV)
      CUROUT = (15.4325*VOUT - 25.33 - EXP((7.Ø86-VOUT)*2.Ø125) -
     1 EXP((6.27-VOUT)*8.45)) * 1.ØE-6 / 2.Ø
      VOUT = VOUT * 1.ØE-3 / 2.Ø
      WRITE(7,11Ø) VOUT, CUROUT
3ØØ   CONTINUE
C
      DO 4ØØ I = I4, NOPNTS
      VOUT = (FLOAT(I-1)*DELTAV)
      CUROUT = (14.1637*VOUT - 125.78/VOUT) * 1.ØE-6 / 2.Ø
      VOUT = VOUT * 1.ØE-3 / 2.Ø
      WRITE(7,11Ø) VOUT, CUROUT
4ØØ   CONTINUE
C
      STOP
      END
```

A.7 Listing of Program GETCHI

This section gives the listing of the preprocessing program GETCHI.

```
FILE: GETCHI   FORTRAN  A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP302

      REAL VOLT2(4096), VOLT3(512), VOLT4(1024)
      REAL CUR2(4096), CUR3(512), CUR4(1024)
      REAL TIME(512), CHI(512)
      COMPLEX CCHI(1024), FACTOR(1024)
      COMMON /GAPINF/ VGAP
C
C     SUBROUTINE GETCHI PRODUCES THE CHI FUNCTION WITH THE
C     APPROPRIATE SPECIFIED DISCRETIZATION.  THIS PROGRAM
C     REQUIRES AS INPUT THE I-V CURVE OF THE SIS JUNCTION.
C     OUTPUT IS TWO-FOLD:
C     (I) THE TABULATED CHI FUNCTION (UNIT 8), AND
C     (II) A COMPATIBLE TABULATED DISCRETIZATION OF THE
C          I-V DATA (UNIT 9).
C
C     THE INPUT I-V DATA WILL BE CHECKED FOR THE FOLLOWING REQUIREMENTS:
C     (A) THE FIRST DATA POINT MUST BE THE ORIGIN;
C     (B) THE I-V FUNCTION MUST BE MONOTONIC;
C     (C) THE RANGE OF INPUT DATA MUST BE SUCH THAT THE VOLTAGE
C         RANGE EXTENDS FROM ZERO TO VOLTAGES WELL INTO THE LINEAR
C         RANGE.  THE INPUT DATA DOES NOT HAVE TO BE EVENLY DISCRETIZED.
C
      WRITE(6,2000)
2000  FORMAT(/1X,'THIS PROGRAM, GETCHI, PERFORMS A PREPROCESSING',
     1/1X,'FUNCTION ON THE INPUT DC I-V CURVE DATA PRIOR TO USING',
     2/1X,'THE PROGRAM SISCAP.  OUTPUT FROM GETCHI IS THE PRE-',
     3/1X,'PROCESSED DC I-V DATA, CHI IN AN APPROPRIATELY DIS-',
     4/1X,'CRETIZED FORM AND VALUES FOR THE PARAMETERS: NOCHI,',
     5/1X,'NOIV, NOPNTS AND RN')
C
      WRITE(6,20)
20    FORMAT(/1X,'INPUT THE GAP VOLTAGE (VGAP) IN VOLTS OF THE SINGLE'
     1/10X,'JUNCTION EQUIVALENT DEVICE (FORMAT E15.8)')
      READ(5,25) VGAP
25    FORMAT(E15.8)
C
      CALL EVEN(VOLT2,CUR2,NUM,DELTAV)
C
      RSTART = 1.0E9
      CALL REFINE(VOLT2,CUR2,NUM,RSTART,RFINIS)
C
      CALL INEXP(VOLT2,CUR2,NUM,VOLT3,CUR3,NOCHI,NOPNTS)
C
      RSTART = RFINIS
      CALL REFINE(VOLT3,CUR3,NOCHI,RSTART,RFINIS)
C
      CALL IVTOCH(VOLT3,CUR3,NOCHI,DELTAV,TIME,CHI,CCHI,FACTOR)
C
      CALL REEXP(VOLT2,CUR2,NUM,VOLT4,CUR4,NOIV,DELTAV)
C
      WRITE(8,10) NOCHI
10    FORMAT(I5)
C
      DO 40 I = 1,NOCHI
40    WRITE(8,30) TIME(I), CHI(I)
30    FORMAT(2E15.8)
```

FILE: GETCHI   FORTRAN A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

```
C
      WRITE(9,1Ø) NOIV
C
      DO 5Ø I = 1,NOIV
5Ø    WRITE(9,3Ø) VOLT4(I), CUR4(I)
C
      WRITE(6,6Ø) NOCHI, NOIV, NOPNTS, RFINIS
6Ø    FORMAT(//1X,'INFORMATION FOR TRANSFERRAL TO PROGRAM SISCAP:'
     1/1X,'OUTPUTTING THE CHI FILE ON UNIT 8: NOCHI = ',I5/
     21X,'OUTPUTTING THE IV FILE ON UNIT 9:  NOIV  = ',I5/
     31X,'NOPNTS = ',I5/
     41X,'RN = ',E15.8,' OHMS')
C
      STOP
      END
      SUBROUTINE EVEN(VOLTV,CURV,NUM,DELTAV)
      REAL CURV(NUM), VOLTV(NUM)
      LOGICAL FLAG
      COMMON /GAPINF/ VGAP
C
C     SUBROUTINE EVEN: THIS SUBROUTINE PROVIDES AN EVENLY SPACED
C     VERSION OF THE EXPERIMENTALLY OBTAINED INPUT I-V DATA.
C     LINEAR INTERPOLATION IS USED TO PPOVIDE THE OUTPUT POINTS
C     THE EXPERIMENTAL DATA IS SCALED TO A SINGLE JUNCTION EQUIVALENT
C
      DELTAV = Ø.ØØØØ5
      NUM = 4Ø96
C
      READ(7,1Ø,END=2ØØØ) NTMP
1Ø    FORMAT(I5)
      WRITE(6,2)
2     FORMAT(/1X,'INPUTTING THE INPUT I-V CURVE DATA FROM UNIT 7')
C
C      WRITE(6,5)
C5     FORMAT(/1X,'SUBROUTINE EVEN: THIS SUBROUTINE PROVIDES AN EVENLY',
C    11X,'SPACED VERSION'/18X,'OF THE EXPERIMENTALLY OBTAINED INPUT I-V
C    2,' DATA'/18X,'LINEAR INTERPOLATION IS USED TO PROVIDE THE POINTS'
C    3/18X,'ON OUTPUT, THE EXPERIMENTAL DATA IS SCALED'/18X,'TO A',
C    4' SINGLE JUNCTION EQUIVALENT')
C
      WRITE(6,2Ø)
2Ø    FORMAT(/1X,'INPUT THE NUMBER OF JUNCTIONS (NOJUN) IN DEVICE (I5)')
      READ(5,3Ø) NOJUN
3Ø    FORMAT(I5)
C
C     ADJUST DELTAV TO CORRESPOND WITH A MULTI-JUNCTION DEVICE
C
      DELTAV = FLOAT(NOJUN) * DELTAV
C
      VOLT = Ø.Ø
      CUR = Ø.Ø
      V1 = Ø.Ø
      C1 = Ø.Ø
      FLAG = .FALSE.
C
```

```
      READ(7,4Ø) V2, C2                                                 GETØ111Ø
4Ø    FORMAT(2E15.8)                                                    GETØ112Ø
C                                                                       GETØ113Ø
      IF(V2.NE.Ø.Ø .OR. C2.NE.Ø.Ø) GO TO 3ØØØ                           GETØ114Ø
C                                                                       GETØ115Ø
      DO 1ØØ I = 1,NUM                                                  GETØ116Ø
      VOLTV(I) = VOLT / FLOAT(NOJUN)                                    GETØ117Ø
      CURV(I) = CUR / FLOAT(NOJUN)                                      GETØ118Ø
      VOLT = VOLT + DELTAV                                              GETØ119Ø
5Ø    IF(VOLT.LT.V2) GO TO 6Ø                                           GETØ12ØØ
      IF(C2.LT.C1 .OR. V2.LT.V1) GO TO 4ØØØ                             GETØ121Ø
      V1 = V2                                                           GETØ122Ø
      C1 = C2                                                           GETØ123Ø
      IF(FLAG) GO TO 7Ø                                                 GETØ124Ø
      READ(7,4Ø,END=7Ø) V2, C2                                          GETØ125Ø
      GO TO 5Ø                                                          GETØ126Ø
6Ø    CUR = C1 + (C2 - C1) / (V2 - V1) * (VOLT - V1)                    GETØ127Ø
      GO TO 1ØØ                                                         GETØ128Ø
7Ø    RTST = V2 / C2                                                    GETØ129Ø
      IF(V2.LT.2.Ø*VGAP*FLOAT(NOJUN)) GO TO 6ØØØ                        GETØ13ØØ
      CUR = VOLT /RTST                                                  GETØ131Ø
      FLAG = .TRUE.                                                     GETØ132Ø
1ØØ   CONTINUE                                                          GETØ133Ø
C                                                                       GETØ134Ø
1ØØØ  RETURN                                                            GETØ135Ø
C                                                                       GETØ136Ø
2ØØØ  WRITE(6,2ØØ1)                                                     GETØ137Ø
2ØØ1  FORMAT(1X,'ERROR DETECTED IN SUBROUTINE EVEN'/                    GETØ138Ø
     11X,'NO DATA FOUND IN IV FILE')                                    GETØ139Ø
      STOP                                                              GETØ14ØØ
3ØØØ  WRITE(6,3ØØ1)                                                     GETØ141Ø
3ØØ1  FORMAT(1X,'ERROR DETECTED IN SUBROUTINE EVEN'/                    GETØ142Ø
     11X,'FIRST POINT IN IV EXPERIMENTAL DATA SHOULD BE Ø.Ø,Ø.Ø')       GETØ143Ø
      STOP                                                              GETØ144Ø
4ØØØ  WRITE(6,4ØØ1) V1, V2, C1, C2                                      GETØ145Ø
4ØØ1  FORMAT(1X,'ERROR DETECTED IN SUBROUTINE EVEN'/                    GETØ146Ø
     11X,'POINTS ARE NOT MONOTONICALLY INCREASING'/                     GETØ147Ø
     21X,'VOLTAGE 1 = ',E15.8,5X,'VOLTAGE 2 = ',E15.8/                  GETØ148Ø
     31X,'CURRENT 1 = ',E15.8,5X,'CURRENT 2 = ',E15.8)                  GETØ149Ø
      STOP                                                              GETØ15ØØ
5ØØØ  WRITE(6,5ØØ1)                                                     GETØ151Ø
5ØØ1  FORMAT(1X,'ERROR DETECTED IN SUBROUTINE EVEN'/                    GETØ152Ø
     11X,'RANGE OF GIVEN I-V DATA POINTS DOES NOT COVER REQD SPACE')    GETØ153Ø
      STOP                                                              GETØ154Ø
6ØØØ  WRITE(6,6ØØ1)                                                     GETØ155Ø
6ØØ1  FORMAT(1X,'ERROR DETECTED IN SUBROUTINE EVEN'/                    GETØ156Ø
     11X,'INPUT I-V CURVE DATA DOES NOT EXTEND FAR ENOUGH '/            GETØ157Ø
     21X,'INTO THE LINEAR RANGE')                                       GETØ158Ø
      STOP                                                              GETØ159Ø
      END                                                               GETØ16ØØ
      SUBROUTINE REFINE(VOL,CUR,NUM,RBEFOR,RAFTER)                      GETØ161Ø
      DIMENSION VOL(NUM), CUR(NUM)                                      GETØ162Ø
C                                                                       GETØ163Ø
C     SUBROUTINE REFINE REMOVES THE OHMIC PORTION FROM THE I-V DATA SUCHGETØ164Ø
C     THAT THE LAST DATA POINT HAS ZERO CURRENT                         GETØ165Ø
```

```
C
C      WRITE(6,9Ø) NUM
C9Ø    FORMAT(/1X,'SUBROUTINE REFINE: REMOVING THE OHMIC PORTION FROM',
C     1/2ØX,'I-V DATA SUCH THAT THE LAST POINT HAS ZERO CURRENT'/
C     22ØX,'NUMBER OF POINTS PROCESSED = ',I5)
C
      GADD = CUR(NUM) / VOL(NUM)
C
      DO 4Ø I = 1,NUM
4Ø    CUR(I) = CUR(I) - GADD * VOL(I)
C
      GBEFOR = 1./RBEFOR
      GAFTER = GBEFOR + GADD
      RAFTER = 1. / GAFTER
C
      RETURN
C
      END
      SUBROUTINE INEXP(TIMIN,CHI,NUM,TOUT,CHOUT,NTOTAL,NOPNTS)
      REAL TOUT(NTOTAL), CHOUT(NTOTAL)
      DIMENSION TIMIN(NUM), CHI(NUM)
      LOGICAL FLAG
      COMMON /GAPINF/ VGAP
      DATA PI, HBAR, E/3.1415926,1.Ø55E-34,1.6Ø2E-19/
      DATA TOL1, TOL2/1.E-7,1.E-3/
C
C     INTERPOLATION PROGRAM
C
C     SUBROUTINE INEXP INTERPOLATES THE I-V DATA SUCH THAT
C     THE SPACING AND NUMBER OF POINTS CORRESPOND TO THE
C     CORRECT SPACING FOR THE CHI CURVE
C
C     THE INTERFOLATION FORMULA USED IS A + B*X + C*EXP(D*X)
C     EQUAL SPACED POINTS MUST BE USED
C
      FLAG = .FALSE.
C
C      WRITE(6,7ØØ)
C7ØØ   FORMAT(/1X,'SUBROUTINE INEXP: INTERPOLATES THE I-V DATA SUCH',
C     11X,'THAT THE SPACING'/19X,'AND NUMBER OF POINTS CORRESPOND'/
C     219X,'TO THE CORRECT SPACING FOR THE CHI CURVE'/)
C
      WRITE(6,72Ø)
72Ø   FCRMAT(/1X,'INPUT THE LO FREQUENCY IN HERTZ (FREQ) (E15.8)')
      READ(5,74Ø) FPEQ
74Ø   FORMAT(E15.8)
      DELTA = E * VGAP / 2.Ø
      N1 = 2.Ø * DELTA / (Ø.Ø64*FREQ*2.Ø*PI*HBAR)
      N2 = 26Ø / N1
      WRITE(6,75Ø) N2, N1
75Ø   FORMAT(/1X,'NUMBER OF LO CYCLES OF CHI IN SISCAP SHOULD BE',
     11X,' > ',I5/
     21X,'THE NUMBER OF POINTS PER LO CYCLE IN SISCAP ',
     3'SHOULD BE > ',I5)
      WRITE(6,755)
```

```
755   FORMAT(/1X,'INPUT NUMBER OF LO CYCLES (NOCYC) (I5)'/1X,'INPUT NO',
     1' OF POINTS PER LO CYCLE (NOPNTS) (I5)')
      READ(5,76Ø) NOCYC, NOPNTS
76Ø   FORMAT(I5)
C
      DELTAT = 1. / FREQ / FLOAT(NOPNTS)
      DELTAW = 1.Ø/FLOAT(NOPNTS)/FLOAT(NOCYC)/DELTAT*2.Ø*PI
      DELTAV = HBAR * DELTAW / E / 2.Ø
      NTOTAL = NOCYC * NOPNTS
      VTOTAL = FLOAT(NTOTAL) * DELTAV
      WRITE(6,554) NTOTAL
554   FORMAT(1X,'NOCHI = ',I5)
C
C      WRITE(6,555) DELTAV, VTOTAL
C555   FORMAT(1X,'DELTAV = ',E15.8,' VOLTS',1X,'TOTAL VOLTAGE REQD = ',
C     1E15.8,' VOLTS')
C
      TIME = Ø.
      IPNT = 2
      VAL = Ø.
      TOUT(1) = TIME
      CHOUT(1) = VAL
C
      NLIMIT = NTOTAL - 1
      DO 1ØØ I = 1,NLIMIT
      TIME = TIME + DELTAV
      IF(FLAG) GO TO 41Ø
6Ø    IF(TIMIN(IPNT+1).GT.TIME) GO TO 7Ø
      IPNT = IPNT + 1
      IF(IPNT+2.GT.NUM) GO TO 41Ø
      GO TO 6Ø
C
7Ø    YM1 = CHI(IPNT-1)
      YØ = CHI(IPNT)
      Y1 = CHI(IPNT+1)
      Y2 = CHI(IPNT+2)
      CURV1 = Y2 - 2.Ø*Y1 + YØ
      CURV2 = Y1 - 2.Ø*YØ + YM1
      IF(CURV1*CURV2 .LT. Ø.Ø) GO TO 47Ø
      IF(ABS(CURV1).LE.TOL1 .OR. ABS(CURV2).LE.TOL1)GOTO47Ø
      D = ALOG(CURV1/CURV2)
      IF(ABS(D).LT.TOL2) GO TO 47Ø
      C = CURV1 / ((EXP(D)-1.Ø)**2)
      A = YØ - C
      B = Y1 - A - C*EXP(D)
      X = (TIME-TIMIN(IPNT))/(TIMIN(IPNT+1)-TIMIN(IPNT))
      VAL = A + B*X + C*EXP(D*X)
      GO TO 4ØØ
C
47Ø    VAL = (TIME-TIMIN(IPNT))/(TIMIN(IPNT+1)-TIMIN(IPNT))
     1 * (CHI(IPNT+1) - CHI(IPNT)) + CHI(IPNT)
C
      GO TO 4ØØ
41Ø   FLAG = .TRUE.
      VAL = Ø.Ø
```

```
FILE: GETCHI   FORTRAN  A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP302

C
400   TOUT(I+1) = TIME
      CHOUT(I+1) = VAL
C
100   CONTINUE
C
      IF(.NOT.FLAG) RETURN
C
      PERCNT = TIMIN(NUM) / VTOTAL * 100.0
      WRITE(6,210) PERCNT
210   FORMAT(1X,'THE GIVEN POINTS WERE NOT SUFFICIENT TO COVER THE'
     1,' REQD INTERP REGION'/1X,'THE FUNCTION WAS ASSUMED',
     2,' TO BE ZERO FOR THE EXTRAPOLATED PORTION',/
     31X,'THE INTERPOLATED REGION EQUALS ',F8.2,'% OF THE TOTAL RANGE')
      RETURN
C
      END
      SUBROUTINE IVTOCH(VOL,CURNEW,NOCHI,DELTAV,TIME,CHI,CCHI,FACTOR)
      REAL VOL(NOCHI), CURNEW(NOCHI), TIME(NOCHI), CHI(NOCHI)
      REAL LENGTH
      COMPLEX CCHI(NOCHI), FACTOR(NOCHI)
C
C     THIS SUBROUTINE TRANSFORMS THE DISCRETIZED I-V DATA
C     INTO THE EQUIVALENT DISCRETIZED CHI DATA
C
C      WRITE(6,300)
C300   FORMAT(/1X,'SUBROUTINE IVTOCH: TRANSFORMING I-V DATA TO CHI DATA'
C
      IBASE = -9
      HBAR = 1.055E-34
      E = 1.602E-19
      NDIV2 = NOCHI
      N = NDIV2 * 2
      IF(N.EQ.512) GO TO 222
      IF(N.EQ.1024) GO TO 223
      IF(N.EQ.2048) GO TO 224
      IF(N.EQ.4096) GO TO 225
      IF(N.EQ.8192) GO TO 226
      WRITE(6,777) N
777   FORMAT(1X,'N DOES NOT EQUAL 512,1024,2048,4096,8192'
     1/1X,'IT EQUALS ',I5)
      WRITE(6,799)
799   FORMAT(1X,'INPUT THE BINARY LEOGARITHM OF THIS NUMBER')
      READ(5,778) IBASE
778   FORMAT(I5)
      GO TO 222
223   IBASE = -10
      GO TO 222
224   IBASE = -11
      GO TO 222
225   IBASE = -12
      GO TO 222
226   IBASE = -13
C
C     SET UP DC VALUE AT ZERO ON I-V CURVE
```

```
C                                                                       GETØ331Ø
222   TMP = CURNEW(1)                                                   GETØ332Ø
      VOL(1) = Ø.                                                       GETØ333Ø
      CCHI(1) = CMPLX(Ø.,Ø.)                                            GETØ334Ø
C                                                                       GETØ335Ø
      DO 1ØØ I = 2,NDIV2                                                GETØ336Ø
      TMP = CURNEW(I)                                                   GETØ337Ø
      DELTAV = VOL(I) - VHLD                                            GETØ338Ø
      VHLD = VOL(I)                                                     GETØ339Ø
      INEXT = N + 2 - I                                                 GETØ34ØØ
      CCHI(INEXT) = CMPLX(Ø.,TMP)                                       GETØ341Ø
1ØØ   CCHI(I) = CMPLX(Ø.,-TMP)                                          GETØ342Ø
11Ø   FORMAT(E15.8,E15.8)                                               GETØ343Ø
      CCHI(NDIV2 + 1) = CMPLX(Ø.,Ø.)                                    GETØ344Ø
C                                                                       GETØ345Ø
C      WRITE(6,446) DELTAV, NDIV2                                       GETØ346Ø
C446   FORMAT(19X,'DELTAV INPUT IS ',E15.8,' VOLTS'/                    GETØ347Ø
C     119X,'NO OF POINTS INPUT = ',I5)                                  GETØ348Ø
      DELTAF = DELTAV / HBAR / 2. / 3.1415926 * E                       GETØ349Ø
      FRMAX = DELTAF * FLOAT(NDIV2)                                     GETØ35ØØ
C                                                                       GETØ351Ø
C     NYQUIST SAMPLING IS TWICE THAT OF MAX FREQUENCY IN W FORM         GETØ352Ø
C                                                                       GETØ353Ø
      DELTAT = 1. / (FRMAX * 2.)                                        GETØ354Ø
      LENGTH = FRMAX * 2.                                               GETØ355Ø
C                                                                       GETØ356Ø
      CALL FFT(IBASE,CCHI,FACTOR)                                       GETØ357Ø
C                                                                       GETØ358Ø
      DO 2ØØ I = 1,NDIV2                                                GETØ359Ø
      TIME(I) = FLOAT(I-1) * DELTAT                                     GETØ36ØØ
C                                                                       GETØ361Ø
C     THE VALUE OF LENGTH/FLOAT(N) EQUAL DELTAF                         GETØ362Ø
C                                                                       GETØ363Ø
      CHI(I) = 2. * REAL(CCHI(I)) * LENGTH / FLOAT(N)                   GETØ364Ø
2ØØ   CONTINUE                                                          GETØ365Ø
C                                                                       GETØ366Ø
C      WRITE(6,31Ø) DELTAT, NDIV2                                       GETØ367Ø
C31Ø   FORMAT(19X,'DELTAT ON OUTPUT = ',E15.8,' SECS'/                  GETØ368Ø
C     119X,'NO OF POINTS OUTPUT = ',I5)                                 GETØ369Ø
C                                                                       GETØ37ØØ
      RETURN                                                            GETØ371Ø
      END                                                               GETØ372Ø
      SUBROUTINE FFT(NUMBER,DATA,FACTOR)                                GETØ373Ø
      INTEGER  GRPSZE,POSN,POINTR,ELMTNO,FACPTR,ABSNUM                  GETØ374Ø
      INTEGER GPPTR                                                     GETØ375Ø
      LOGICAL  INVERT                                                   GETØ376Ø
      COMPLEX  TEMP,W,FACTOR(NUMBER),DATA(NUMBER),WSCALE,TEMP1,         GETØ377Ø
     1TEMP2                                                             GETØ378Ø
      DATA  PI/3.1415926/                                               GETØ379Ø
C                                                                       GETØ38ØØ
C  AUTHOR: R.G. HICKS, UNIV. OF QLD., APRIL 1977.                       GETØ381Ø
C                                                                       GETØ382Ø
C     AIM: THIS SUBROUTINE PERFORMS A FAST FOURIER TRANSFORM ON THE     GETØ383Ø
C           INPUT DATA USING THE COOLEY-TURKEY ALGORITHM                GETØ384Ø
C           THERE MUST BE 2**NUMBER POINTS GIVEN                        GETØ385Ø
```

```
C           AND THE GIVEN POINTS ARE DESTROYED BY THE SUBROUTINE.
C           IF NUMBER IS NEGATIVE, AN INVERSE FOURIER TRANSFORM IS DONE.
C
      INVERT = .FALSE.
      ABSNUM = NUMBER
      IF(NUMBER.GT.Ø) GO TO 1Ø
      ABSNUM = -NUMBER
      INVERT = .TRUE.
1Ø    N = 2 ** ABSNUM
      ARG = 2. * PI / FLOAT(N)
      TEMP = CMPLX(Ø.,ARG)
      W = CEXP(TEMP)
C
      FACTOR(1) = W
      NDIV2 = N/2
      DO 15 INDEX = 2,NDIV2
15    FACTOR(INDEX) = FACTOR(INDEX - 1)*W
C
      IF(INVERT) GO TO 2Ø
      DO 17 POSN = 1,N
17    DATA(POSN) = CONJG(DATA(POSN))
C
2Ø    GRPSZE = N
      NOGRP = 1
      ITERNO = Ø
25    LIMIT = GRPSZE/2
      ITERNO = ITERNO + 1
      POINTR = GRPSZE/2 + 1
      POSN = 1
C
      DO 6Ø GRPTR = 1, NOGRP
      DO 5Ø ELMTNO = 1,LIMIT
      IPOSN = POSN - 1
      CALL FACNO(IPOSN,N,ITERNO,FACPTR)
      IF(FACPTR.EQ.Ø) GO TO 3Ø
      WSCALE = FACTOR(FACPTR)
      TEMP1 = DATA(POINTR) * WSCALE + DATA(POSN)
      WSCALE = -WSCALE
      TEMP2 = DATA(POINTR) * WSCALE + DATA(POSN)
      GO TO 4Ø
3Ø    TEMP1 = DATA(POINTR) + DATA(POSN)
      TEMP2 = DATA(POSN) - DATA(POINTR)
4Ø    DATA(POSN) = TEMP1
      DATA(POINTR) = TEMP2
      POSN = POSN + 1
5Ø    POINTR = POINTR + 1
C
      POSN = POSN + GRPSZE/2
6Ø    POINTR = POINTR + GRPSZE/2
C
      GRPSZE = GRPSZE/2
      NOGRP = NOGRP * 2
      IF(GRPSZE .GT. 1) GO TO 25
C
      DO 7Ø POSN = 1,N
      IPOSN = POSN - 1
      CALL JUXTA(N,IPOSN,JUXPOS)
      JUXPOS = JUXPOS + 1
      IF(JUXPOS.LT.POSN) GO TO 70
      TEMP = DATA(POSN)
      DATA(POSN) = DATA(JUXPOS)
      DATA(JUXPOS) = TEMP
70    CONTINUE
C
      IF(INVERT) RETURN
      DO 80 POSN = 1,N
      TEMP = CONJG(DATA(POSN))
80    DATA(POSN) = TEMP / N
C
      RETURN
      END
      SUBROUTINE JUXTA(N,IPOSN,JUXPOS)
C
C     THIS SUBROUTINE DETERMINES THE CORRECT MULTIPLICATION SEQUENCING
C     IN THE BUTTERFLY ALGORITHM OF THE FAST FOURIER TRANSFORM
C
      NN = N
      NUM = IPOSN
      JUXPOS = 0
C
      DO 10 I=1,N
      NN = NN/2
      MEM = NUM
      NUM = NUM/2
      IREM = MEM - 2*NUM
      JUXPOS = 2*JUXPOS + IREM
      IF(NN.EQ.1) GO TO 20
10    CONTINUE
20    RETURN
      END
      SUBROUTINE FACNO(IPOSN,N,ITERNO,FACPTR)
      INTEGER FACPTR
C
C     THIS SUBROUTINE DETERMINES THE CORRECT FACTOR FOR CORRECT
C     CALCULATION OF THE BUTTERFLY ALGORITHM IN THE FAST FOURIER TRANSFO
C
      FACPTR = 0
      IDIV = N/(2**ITERNO)
      ITEMP = IPOSN
      ITEMP = ITEMP/IDIV
      MUL = N/2
      DO 10 I=1,ITERNO
      IREM = ITEMP - ITEMP/2*2
      FACPTR = IREM*MUL + FACPTR
      ITEMP = ITEMP/2
      MUL = MUL/2
10    CONTINUE
      RETURN
      END
      SUBROUTINE REEXP(TIMIN,CHI,NUM,TNEW,CHNEW,NOIV,DELTAV)
      REAL TNEW(NOIV), CHNEW(NOIV)
      DIMENSION TIMIN(NUM), CHI(NUM)
      LOGICAL FLAG
      COMMON /GAPINF/ VGAP
      DATA TOL1, TOL2/1.E-7,1.E-3/
C
C     INTERPOLATION PROGRAM
C
C     THIS SUBROUTINE INTERPOLATES THE DATA SUCH THAT THE VECTOR
C     J MAY BE CALCULATED WITH REASONABLE ACCURACY.
C     TYPICALLY DELTAV SHOULD BE (VGAP/3.ØE-3)*2.ØE-5 VOLTS
C     SHOULD BE USED.
C
C     THE INTERPOLATION FORMULA USED IS A + B*X + C*EXP(D*X)
C
C     THE POINTS USED MUST BE EQUALLY SPACED
C
      FLAG = .FALSE.
      VSUG = (VGAP/3.ØE-3) * Ø.ØØØØ2
C
C
C      WRITE(6,3Ø)
C3Ø    FORMAT(/1X,'SUBROUTINE REEXP: PROVIDES THE OUTPUT IV DATA',
C      1,' FOR PROGRAM SISCAP')
      WRITE(6,4Ø) VSUG
4Ø    FORMAT(/1X,'INPUT DELTAV (TYP ',E11.4,' VOLTS) (E15.8)'/
     11X,'INPUT NUMBER OF POINTS REQUIRED (NOIV) (PWR OF 2) (TYP 1Ø24)',
     2' (I5)')
      READ(5,5Ø) DELTAV, NOIV
5Ø    FORMAT(E15.8/I5)
C
      NTOTAL = NOIV
      VTOTAL = FLOAT(NTOTAL) * DELTAV
C
      TIME = Ø.
      IPNT = 2
      VAL = Ø.
      TNEW(1) = TIME
      CHNEW(1) = VAL
C
      NLIMIT = NTOTAL - 1
      DO 1ØØ I = 1,NLIMIT
      TIME = TIME + DELTAV
      IF(FLAG) GO TO 41Ø
6Ø    IF(TIMIN(IPNT+1).GT.TIME) GO TO 7Ø
      IPNT = IPNT + 1
      IF(IPNT+2.GT.NUM) GO TO 41Ø
      GO TO 6Ø
C
7Ø    YM1 = CHI(IPNT-1)
      YØ = CHI(IPNT)
      Y1 = CHI(IPNT+1)
      Y2 = CHI(IPNT+2)
      CURV1 = Y2 - 2.Ø*Y1 + YØ
      CURV2 = Y1 - 2.Ø*YØ + YM1
```

FILE: GETCHI   FORTRAN A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

```
      IF(CURV1*CURV2 .LT. Ø.Ø) GO TO 47Ø
      IF(ABS(CURV1).LE.TOL1 .OR. ABS(CURV2).LE.TOL1)GOTO47Ø
      D = ALOG(CURV1/CURV2)
      IF(ABS(D).LT.TOL2) GO TO 47Ø
      C = CURV1 / ((EXP(D)-1.Ø)**2)
      A = YØ - C
      B = Y1 - A - C*EXP(D)
      X = (TIME-TIMIN(IPNT))/(TIMIN(IPNT+1)-TIMIN(IPNT))
      VAL = A + B*X + C*EXP(D*X)
      GO TO 4ØØ
C
47Ø    VAL = (TIME-TIMIN(IPNT))/(TIMIN(IPNT+1)-TIMIN(IPNT))
     1 * (CHI(IPNT+1) - CHI(IPNT)) + CHI(IPNT)
C
      GO TO 4ØØ
41Ø   FLAG = .TRUE.
      VAL = Ø.Ø
C
4ØØ   TNEW(I+1) = TIME
      CHNEW(I+1) = VAL
1ØØ   CONTINUE
C
      IF(.NOT.FLAG) RETURN
C
      PERCNT = TIMIN(NUM) / VTOTAL * 1ØØ.Ø
      WRITE(6,21Ø) PERCNT
21Ø   FORMAT(1X,'THE GIVEN POINTS WERE NOT SUFFICIENT TO COVER THE'
     1,' REQD INTERP REGION'/1X,'THE FUNCTION WAS ASSUMED',
     2,' TO BE ZERO FOR THE EXTRAPOLATED PORTION',/
     31X,'THE INTERPOLATED REGION EQUALS ',F8.2,'% OF THE TOTAL RANGE')
      RETURN
C
      END
```

A.8 Listing of Program SISCAP

This section gives the listing of the SIS mixer analysis program SISCAP.

```
FILE: SISCAP   FORTRAN  A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP302

      COMPLEX ZIMPED(128), VSOURC(128)
      COMPLEX CVAL(128), FACTOR(2048), J(1024), ZSSB(127)
      COMPLEX Y0, Y1, Y1M, FIDDLE, ZLO, CHOLD
      REAL VOLT(128), CHI(512), CUR(128), OLDVOL(128)
      REAL IVDAT(1024)
      LOGICAL FLAG
      COMMON /PERIPH/ IN, IOUT, IPLOT
      COMMON /ERRALL/ TOL, EPS
      COMMON /RBOUT/ VDC, VLO, VJDC, VJLO, CURDC, CURPMP, RLOSS, RTEMP
      DATA E/1.602E-19/, HBAR/1.055E-34/, PI/3.1415926/
C
      IN = 5
      IOUT = 6
      IPLOT = 20
C
C     OBTAIN THE VARIOUS INPUT PARAMETERS, BOTH JUNCTION AND ALGORITHM
C     PARAMETERS
C
      READ(11,1) ISIG, NODAT, DELV, VSTART, VLOSTR, DELVLO, NLO, TOL,
     1 EPS, NOPNTS, IBASE1, NOIV, IBASE2, NOCHI, NOPER, MMAX, NMAX,
     2 RN, CAP, TEMP, FREQ, FIF, P, RID, Z0
1     FORMAT(I5/I5,4(/E15.8),/I5,2(/E15.8),8(/I5),8(/E15.8))
C
      NOCOL = 4
C
C      WRITE(IPLOT,4001) NODAT, NOCOL
C4001  FORMAT(2I5)
C
      WRITE(6,700) FREQ,FIF,CAP,RN,TEMP,IBASE2,NOIV,IBASE1,NOPNTS,MMAX,
     1NMAX, ISIG, NOPER, NOCHI, NODAT, TOL, EPS, DELV, VSTART,
     2VLOSTR, DELVLO, NLO, P, RID, Z0
700   FORMAT(/1X,'FREQ = ',E15.8,'HZ',1X,'FIF = ',E15.8,'HZ'/
     11X,'CAP  = ',E15.8,'F ', 1X,'RN  = ',E15.8,'OHMS'/
     21X,'TEMP = ',E15.8,'K '/
     31X,'IBASE2 = ',I5,1X,'NOIV   = ',I5/
     41X,'IBASE1 = ',I5,1X,'NOPNTS = ',I5/
     51X,'MMAX   = ',I5,1X,'NMAX   = ',I5/
     61X,'ISIG   = ',I5,1X,'NOPER  = ',I5/
     71X,'NOCHI  = ',I5,1X,'NODAT  = ',I5/
     81X,'TOL = ',E15.8,9X,'EPS = ',E15.8/
     91X,'DELV = ',E15.8,'VOLTS',3X,'VSTART = ',E15.8,'VOLTS'/
     11X,'VLOSTR = ',E15.8,'VOLTS',1X,'DELVLO = ',E15.8,'VOLTS'/
     11X,'NLO = ',I5,19X,'P = ',E15.8/
     21X,'RID = ',E15.8,'OHMS',5X,'Z0 = ',E15.8,'OHMS')
C
C     CHECK THE VALIDITY OF CONTROL PARAMETERS
C
      IF(FREQ.LE.FIF) GO TO 8002
      IF(FIF.LE.0 0) GO TO 8002
      IF(2**IBASE2.NE.NCIV) GO TO 8002
      IF(2**IBASE1.NE.NOPNTS) GO TO 8002
      IF(MMAX.GT.NOPNTS/2-1) GO TO 8002
      IF(NMAX.GT.NOPNTS/2-1) GO TO 8002
      IF(NOCHI.LT.NOPNTS) GO TO 8002
      IF(ISIG.LE.0.OR.ISIG.GT.3) GO TO 8002
```

```
C
C     OBTAIN THE CHI FUNCTION BY READING IT FROM DEVICE 4
C
      REWIND 4
      WRITE(6,6Ø1)
6Ø1   FORMAT(/1X,'INPUTTING THE CHI FUNCTION')
      READ(4,5Ø) NFINAL
5Ø    FORMAT(I5)
      IF(NFINAL.NE.NOCHI) GO TO 8ØØØ
      DUMHLD = Ø.Ø
      DO 7Ø I = 1,NFINAL
      READ(4,8Ø) TMP, CHI(I)
8Ø    FORMAT(2E15.8)
      DELTAT = TMP - DUMHLD
      DUMHLD = TMP
7Ø    CONTINUE
      WRITE(6,6Ø2) DELTAT, NFINAL
6Ø2   FORMAT(1X,'DELTAT = ',E15.8,' SECS'/
     11X,'NUMBEP OF POINTS READ = ',I5)
C
C     OBTAIN THE SIS I-V CHARACTERISTIC BY READING IT FROM DEVICE 8
C
      REWIND 8
498   WRITE(6,5Ø1)
5Ø1   FORMAT(/1X,'INPUTTING THE SIS IV CHARACTERISTIC')
      READ(8,1ØØ) NDUM
1ØØ   FORMAT(I5)
      IF(NDUM.NE.NOIV) GO TO 7ØØØ
      DUMHLD = Ø.Ø
      DO 11Ø II = 1,NDUM
      READ(8,12Ø) DUMMY, IVDAT(II)
12Ø   FORMAT(2E15.8)
      DELTAV = DUMMY - DUMHLD
      DUMHLD = DUMMY
11Ø   CONTINUE
      WRITE(6,5Ø7) DELTAV, NDUM
5Ø7   FORMAT(1X,'DELTAV = ',E15.8,' VOLTS'/
     11X,'NUMBEP OF POINTS READ = ',I5)
C
C     CALCULATE THE VECTOR J
C
      WRITE(6,5Ø2)
5Ø2   FORMAT(/1X,'CALCULATING THE VECTOR J')
      CALL GETJ(IVDAT,NOIV,FREQ,IBASE2,J,FACTOR,RN,DELTAV)
C
C     INITIALIZE TO ZERO THE JUNCTION VOLTAGE VECTOR
C
      DO 1Ø I = 1,NOPNTS
      OLDVOL(I) = Ø.
1Ø    VOLT(I) = Ø.
C
C     INPUT THE HARMONIC IMPEDANCES FROM UNIT NO. 15
C
      WRITE(6,6Ø3)
6Ø3   FORMAT(/1X,'INPUTTING THE LO HARMONIC IMPEDANCES')
```

```
      CALL GETZ(ZIMPED,NOPNTS,FREQ,CAP,VSOURC)
      ZLO = ZIMPED(2)
C
C     GET THE SIDEBAND IMPEDANCES FROM UNIT 14
C
      WRITE(6,499)
499   FORMAT(/1X,'INPUTTING SIDEBAND TERMINATION INFORMATION')
      CALL GETSSB(ZSSB,NOPNTS,FREQ,FIF,CAP)
      YØ = CMPLX(1.Ø,Ø.Ø) / ZSSB(1)
      Y1 = CMPLX(1.Ø,Ø.Ø) / ZSSB(2)
      Y1M = CMPLX(1.Ø,Ø.Ø) / CONJG(ZSSB(3))
C
      VLO = VLOSTR - DELVLO
      DO 5ØØØ LIM = 1,NLO
      VLO = VLO + DELVLO
C
      VDC = VSTART - DELV
C
C     RUN THE LARGE SIGNAL ANALYSIS PROGRAM 'NODAT' TIMES
C     FOR EACH DC VOLTAGE BIAS POINT TO BE CONSIDERED
C
      DO 5ØØØ KIM = 1,NODAT
      KKIMM = KIM + (LIM - 1) * NODAT
      WRITE(6,6ØØØ) KKIMM
6ØØØ  FORMAT(/1X,'***********************************'/
     11X,'*********CASE ',I3,'***************'/
     21X,'***********************************'/)
      VDC = VDC + DELV
C
      WRITE(6,6Ø7) VDC, VLO
6Ø7   FORMAT(/1X,'EMBEDDING THEVENIN VDC SOURCE',8X,'= ',E15.8,' VOLTS'/
     11X,'EMBEDDING THEVENIN VLO SOURCE (PEAK) = ',E15.8,' VOLTS')
C
      VJDC = Ø.Ø
      VJLO = Ø.Ø
      CURDC = Ø.Ø
      CURPMP = Ø.Ø
      RLOSS = Ø.Ø
      RTEMP = Ø.Ø
C
C     DETERMINE WHICH METHOD OF ANALYSIS IS REQUIRED
C
C     1 = THREE FREQUENCY MODEL ASSUMPTION
C     2 = VOLTAGE UPDATE GENERAL ANALYSIS
C     3 = MULTIPLE REFLECTION GENERAL ANALYSIS TECHNIQUE
C
      GO TO (9ØØØ,4ØØØ,4ØØØ), ISIG
C
C     FOR THE 3-PORT ANALYSIS METHOD
C
9ØØØ  VJDC = VDC
      VJLO = VLO
C
      ARG = 2.Ø * PI / FLOAT(NOPNTS)
      DO 7777 MNZ = 1,NOPNTS
```

```
FILE: SISCAP   FORTRAN  A   NASA/GISS CONVEPSATIONAL MONITOR SYSTEM SP3Ø2

      PHASE = ARG * FLOAT(MNZ -1)                                       SISØ166Ø
7777  VOLT(MNZ) = VDC - VLO * COS(PHASE)                                SISØ167Ø
C                                                                       SISØ168Ø
      WRITE(6,8888)                                                     SISØ169Ø
8888  FORMAT(/1X,'PERFORMING A 3-PORT ANALYSIS METHOD')                 SISØ17ØØ
      GO TO 3ØØØ                                                        SISØ171Ø
C                                                                       SISØ172Ø
C     FOR THE GENERAL ANALYSIS METHOD                                   SISØ173Ø
C                                                                       SISØ174Ø
4ØØØ  VSOURC(1) = VDC                                                   SISØ175Ø
      VSOURC(2) = VLO/2.                                                SISØ176Ø
C                                                                       SISØ177Ø
C     BCAP = 2. * PI * FREQ * CAP                                       SISØ178Ø
C     FIDDLE = ZLO * CMPLX(Ø.,BCAP) + CMPLX(1.Ø,Ø.Ø)                    SISØ179Ø
C     VSOURC(2) = VSOURC(2) / FIDDLE                                    SISØ18ØØ
C                                                                       SISØ181Ø
      RSOURC = REAL(ZLO)                                                SISØ182Ø
      PLO = (CABS(VSOURC(2))**2) / 2. / RSOURC                          SISØ183Ø
      WRITE(6,4ØØ2) PLO                                                 SISØ184Ø
4ØØ2  FORMAT(/1X,'INCIDENT LOCAL OSCILLATOR POWER = ',E15.8,' WATTS'/)  SISØ185Ø
C                                                                       SISØ186Ø
      IF(ISIG.EQ.2) GO TO 6ØØ3                                          SISØ187Ø
C                                                                       SISØ188Ø
C     FOR THE MULTIPLE REFLECTION TECHNIQUE                             SISØ189Ø
C                                                                       SISØ19ØØ
      CALL MRT(VOLT,CUR,NOPNTS,IBASE1,CHI,NOCHI,RN,                     SISØ191Ø
     1FACTOR,FREQ,ZIMPED,VSOURC,OLDVOL,FLAG,ZØ,NOPER)                   SISØ192Ø
C                                                                       SISØ193Ø
      IF(FLAG) GO TO 5ØØ1                                               SISØ194Ø
C                                                                       SISØ195Ø
      DO 4321 MGM = 1,NOPNTS                                            SISØ196Ø
4321  WRITE(6,4322) MGM, VOLT(MGM), CUR(MGM)                            SISØ197Ø
4322  FORMAT(1X,'POINT NO. = ',I5,'VOLTAGE = ',E15.8,'CURRENT = ',E15.8)SISØ198Ø
C                                                                       SISØ199Ø
      GO TO 6ØØ1                                                        SISØ2ØØØ
C                                                                       SISØ2Ø1Ø
C     FOR THE VOLTAGE UPDATE METHOD                                     SISØ2Ø2Ø
C                                                                       SISØ2Ø3Ø
6ØØ3  WRITE(6,6Ø4)                                                      SISØ2Ø4Ø
6Ø4   FORMAT(/1X,'GENERAL NONLINEAR ANALYSIS USING VOLTAGE UPDATE',     SISØ2Ø5Ø
     1' METHOD'/1X,'BEGINNING THE ITERATION PHASE')                     SISØ2Ø6Ø
C                                                                       SISØ2Ø7Ø
      DO 1ØØØ I = 1,NOPER                                               SISØ2Ø8Ø
C                                                                       SISØ2Ø9Ø
      WRITE(6,4ØØ) I                                                    SISØ21ØØ
4ØØ   FORMAT(/8X,'ITERATION NUMBER = ',I5)                              SISØ211Ø
C                                                                       SISØ212Ø
      CALL SISNL(VOLT,CUR,NOPNTS,IBASE1,CHI,NOCHI,RN,FACTOR,FREQ,RID)   SISØ213Ø
      CALL LINEAR(VOLT,CUR,NOPNTS,IBASE1,FACTOR,ZIMPED,VSOURC,RID)      SISØ214Ø
      CALL COMPAR(VOLT,CLDVOL,NOPNTS,FLAG)                              SISØ215Ø
C                                                                       SISØ216Ø
      DO 5ØØ JJ = 1,NOPNTS                                              SISØ217Ø
5ØØ   OLDVOL(JJ) = P * VOLT(JJ) + (1.Ø - P) * OLDVOL(JJ)                SISØ218Ø
C                                                                       SISØ219Ø
      IF(.NCT. FLAG) GO TO 2ØØØ                                         SISØ22ØØ
```

```
FILE: SISCAP   FORTRAN  A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

C
1ØØØ  CONTINUE
C
      WRITE(6,6Ø6) NOPER
6Ø6   FORMAT(/8X,'LARGE SIGNAL ANALYSIS DID NOT CONVERGE IN ',I5,'ITNS')
      GO TO 5ØØ1
C
2ØØØ  WRITE(6,2ØØ2) I
2ØØ2  FORMAT(/1X,'LARGE SIGNAL ANALYSIS COMPLETED IN ',I5,' ITNS'/)
C
C     CORRECT FOR THE IDENTITY ELEMENT IN THE VOLTAGE
C     UPDATE METHOD
C
6ØØ2  DO 9ØØ2 IM = 1,NOPNTS
9ØØ2  CUR(IM) = CUR(IM) + VOLT(IM) / RID
C
      DO 4323 MGM = 1,NOPNTS
4323  WRITE(6,4322) MGM, VOLT(MGM), CUR(MGM)
C
C     DETERMINE THE NONLINEAR OPERATING POINT PARAMETERS
C     FROM THE RESULTS OF THE GENERAL ANALYSIS
C
6ØØ1  DO 2Ø1Ø IM = 1,NOPNTS
      CVAL(IM) = VOLT(IM)
2Ø1Ø  CONTINUE
      CALL FFT(IBASE1,CVAL,FACTOR)
      VJDC = REAL(CVAL(1))
      VJLO = CABS(CVAL(2)) * 2.Ø
C
      VBIAS = VJDC
      VNORM = E * VBIAS / HBAR
      CALL EVALJ(VNORM,CHOLD,J,NOIV,DELTAV,RN)
      CURDC = AIMAG(CHOLD)
C
C     SMALL SIGNAL ANALYSIS SECTION
C
C     DO THE SMALL SIGNAL CONVERSION AND NOISE ANALYSIS
C
3ØØØ  CALL TOPS(FREQ,FIF,IBASE2,NOIV,IBASE1,NOPNTS,MMAX,CAP,RN,TEMP,
     1NMAX,VOLT,YØ,Y1,Y1M,ZLO,IVDAT,J,FACTOR,DELTAV,ISIG,ZSSB)
C
C5ØØ1  WRITE(IPLOT,45ØØ) VJDC, CURDC, CURPMP, RLOSS, RTEMP
C45ØØ  FORMAT(5E15.8)
C
5ØØ1  CONTINUE
C
5ØØØ  CONTINUE
C
      STOP
C
7ØØØ  WRITE(6,1ØØ1)
1ØØ1  FORMAT(1X,'PROBLEM WITH NUMBER OF POINTS IN IVDAT FILE')
      STOP
C
8ØØØ  WRITE(6,8ØØ1)
```

FILE: SISCAP   FORTRAN A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

```
8ØØ1  FORMAT(1X,'PROBLEM WITH NUMBER OF POINTS IN NOCHI FILE')          SISØ276Ø
      STOP                                                              SISØ277Ø
C                                                                       SISØ278Ø
8ØØ2  WRITE(6,8ØØ3)                                                     SISØ279Ø
8ØØ3  FORMAT(1X,'ERROR DETECTED IN CONTROL PARAMETERS')                 SISØ28ØØ
      STOP                                                              SISØ281Ø
C                                                                       SISØ282Ø
      END                                                               SISØ283Ø
      SUBROUTINE GETSSB(ZSSB,NOPNTS,FREQ,FIF,CAP)                       SISØ284Ø
      COMPLEX ZSSB(NOPNTS), YCAP                                        SISØ285Ø
      COMPLEX YSIG, YIM                                                 SISØ286Ø
      DATA PI/3.1415926/                                                SISØ287Ø
      DATA NOPRT/1Ø/                                                    SISØ288Ø
C                                                                       SISØ289Ø
C     READ IN THE SIDEBAND IMPEDANCES FROM UNIT NUMBER                  SISØ29ØØ
C     14.  THE ORDER OF THE IMPEDANCE DATA IS IF,                       SISØ291Ø
C     USB NO.1, LSB NO.1, USB NO.2, LSB NO.2, AND SO ON                 SISØ292Ø
C                                                                       SISØ293Ø
      READ(14,11Ø1) NCHK                                                SISØ294Ø
11Ø1  FORMAT(I5)                                                        SISØ295Ø
C     IF(NCHK.NE.NOPNTS-1) GO TO 8ØØØ                                   SISØ296Ø
      NOMIN1 = NOPNTS - 1                                               SISØ297Ø
      READ(14,112Ø) ZSSB(1)                                             SISØ298Ø
112Ø  FORMAT(2E15.8)                                                    SISØ299Ø
      WRITE(6,3ØØ) NOPRT, ZSSB(1)                                       SISØ3ØØØ
3ØØ   FORMAT(1X,'THE FIRST ',I3,' SIDEBAND IMPEDANCES (EXC CAPAC): ',   SISØ3Ø1Ø
     1/1X,'    Ø ',E15.8,'+J',E15.8,'OHMS')                             SISØ3Ø2Ø
      DO 8Ø1 II = 2,NOMIN1                                              SISØ3Ø3Ø
      ZSSB(II) = CMPLX(Ø.Ø,Ø.Ø)                                         SISØ3Ø4Ø
      ISDBD = II / 2                                                    SISØ3Ø5Ø
      IF(ISDBD*2.NE.II .AND. II.NE.1) ISDBD = -ISDBD                    SISØ3Ø6Ø
      IF(II.GT.NCHK) GO TO 2ØØØ                                         SISØ3Ø7Ø
      READ(14,112Ø) ZSSB(II)                                            SISØ3Ø8Ø
2ØØØ  IF(II.LE.NOPRT) WRITE(6,3Ø1) ISDBD, ZSSB(II)                      SISØ3Ø9Ø
3Ø1   FORMAT(1X,I5,1X,E15.8,'+J',E15.8,'OHMS')                          SISØ31ØØ
      IF(II/2*2.NE.II) ZSSB(II) = CONJG(ZSSB(II))                       SISØ311Ø
8Ø1   CONTINUE                                                          SISØ312Ø
C                                                                       SISØ313Ø
C     ADD IN THE JUNCTION PARASITIC CAPACITANCE CONTRIBUTION            SISØ314Ø
C                                                                       SISØ315Ø
      CON = 2. * PI * CAP                                               SISØ316Ø
      DO 398 II = 1,NCHK                                                SISØ317Ø
      IF(CABS(ZSSB(II)) .LT. 1.E-1Ø) GO TO 398                          SISØ318Ø
      M = II / 2                                                        SISØ319Ø
      IF(M.EQ.1 .OR. M.EQ.Ø) GO TO 398                                  SISØ32ØØ
      IF(M*2.NE.II.AND.II.NE.1) M = -M                                  SISØ321Ø
      BCAP = (FLOAT(M)*FREQ + FIF) * CON                                SISØ322Ø
      YCAP = CMPLX(Ø.,BCAP)                                             SISØ323Ø
      ZSSB(II) = 1./(1./ZSSB(II) + YCAP)                                SISØ324Ø
398   CONTINUE                                                          SISØ325Ø
C                                                                       SISØ326Ø
      YSIG = CMPLX(1.Ø,Ø.Ø) / ZSSB(2)                                   SISØ327Ø
      YIM = CMPLX(1.Ø,Ø.Ø) / ZSSB(3)                                    SISØ328Ø
      WRITE(6,7ØØØ) YSIG, YIM                                           SISØ329Ø
7ØØØ  FORMAT(1X,'Y SIGNAL (INCL CAPAC) = ',E15.8,' +J ',E15.8,'S'/1X,   SISØ33ØØ
FILE: SISCAP   FORTRAN  A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

     1'Y IMAGE  (INCL CAPAC) = ',E15.8,' +J ',E15.8,'S')
C
      WRITE(6,2ØØ) NCHK
2ØØ   FORMAT(1X,'NUMBER OF POINTS READ = ',I5)
      RETURN
C
C8ØØØ  WRITE(6,8ØØ1)
C8ØØ1  FORMAT(1X,'PROBEM WITH NUMBER OF POINTS IN ZSSB FILE')
C      STOP
      END
      SUBROUTINE GETZ(ZIMPED,NOPNTS,FREQ,CAP,VSOURC)
      COMPLEX ZIMPED(NOPNTS), ZCAP, VSOURC(NOPNTS)
      COMPLEX YPMP
      DATA PI/3.1415926/
      DATA NOPRT /1Ø/
C
C     READ IN FROM UNIT NO. 15 THE EMBEDDING IMPEDANCES
C     AT THE LO AND ITS HARMONICS
C
      CONST = 2. * PI * FREQ * CAP
      NODIV2 = NOPNTS/2
C
      READ(15,1ØØ) NCHK
1ØØ   FORMAT(I5)
C     IF(NCHK.NE.NODIV2) GO TO 1ØØØ
C
      READ(15,15Ø) ZIMPED(1)
      WRITE(6,3ØØ) NOPRT, ZIMPED(1)
3ØØ   FORMAT(1X,'THE FIRST ',I3,' LO HARM IMPEDANCES (EXC CAPAC) ARE: '
     1/1X,'    Ø ',E15.8,'+J',E15.8,'OHMS')
      DO 1Ø I = 2,NODIV2
      ZIMPED(I) = CMPLX(Ø.Ø,Ø.Ø)
      IMIN1 = I - 1
      IF(I.GT.NCHK) GO TO 2ØØØ
      READ(15,15Ø) ZIMPED(I)
 15Ø  FORMAT(2E15.8)
 2ØØØ IF(I.LE.NOPRT) WRITE(6,3Ø1) IMIN1, ZIMPED(I)
3Ø1   FORMAT(1X,I5,1X,E15.8,'+J',E15.8,'OHMS')
      IF(I.EQ.2) GO TO 1Ø
C
C     ADD IN THE JUNCTION PARASITIC CAPACITANCE CONTRIBUTION
C
      XCAP = -1. / CONST / FLOAT(I-1)
      ZCAP = CMPLX(Ø.,XCAP)
      IF(CABS(ZIMPED(I)).LT.1.ØE-1Ø) GO TO 1Ø
      ZIMPED(I) = 1. / (1./ZIMPED(I) + 1./ZCAP)
1Ø    CONTINUE
      YPMP = CMPLX(1.Ø,Ø.Ø) / ZIMPED(2)
      WRITE(6,11Ø) YPMP
11Ø   FORMAT(1X,'Y PUMP (INCL CAPAC) = ',E15.8,' +J ',E15.8,'S')
C
C     ZERO OUT THE SOURCE VOLTAGE VECTOR
C
      DO 2Ø I = 1,NODIV2
2Ø    VSOURC(I) = CMPLX(Ø.,Ø.)
```

```
C
      WRITE(6,200) NCHK
200   FORMAT(1X,'NUMBER OF POINTS READ = ',I5)
      RETURN
C
C1000  WRITE(6,1001)
C1001  FORMAT(1X,'ERROR IN NUMBER OF POINTS IN ZLO FILE')
C
C     STOP
C
      END
      SUBROUTINE LINEAR(VOLT,CUR,NOPNTS,IBASE,FACTOR,ZIMPED,VSOURC,RID)
      REAL VOLT(NOPNTS), CUR(NOPNTS)
      COMPLEX FACTOR(NOPNTS), ZIMPED(NOPNTS), CTEMP(128), VSOURC(NOPNTS)
      COMMON /RBOUT/ VDC, VLO, VJDC, VJLO, CURDC, CURPMP, RLOSS, RTEMP
C
C     THIS IS THE LINEAR EMBEDDING NETWORK CALCULATION PART
C     OF THE VOLTAGE UPDATE METHOD
C
      DO 10 I = 1,NOPNTS
10    CTEMP(I) = -CUR(I)
C
      CALL FFT(IBASE,CTEMP,FACTOR)
C
C     OBTAIN THE PUMPED I-V CURVE CURRENT VALUE ON THE WAY THROUGH
C
      CURPMP = -REAL(CTEMP(1))
C
C     DON'T FORGET TO TAKE INTO ACCOUNT THE EFFECT OF
C     THE IDENTITY ELEMENT
C
      NODIV2 = NOPNTS/2
      DO 20 I = 1,NODIV2
      CTEMP(I) = CTEMP(I)*(ZIMPED(I)*RID)/(ZIMPED(I)+RID) +
     1VSOURC(I) * RID / (ZIMPED(I) + RID)
      IF(I.EQ.1) GO TO 20
      ICORR = NOPNTS - I + 2
      CTEMP(ICORR) = CONJG(CTEMP(I))
20    CONTINUE
      CTEMP(NODIV2+1) = CMPLX(0.,0.)
C
      CALL ZERO(CTEMP,NOPNTS)
C
C     COMPENSATE FOR THE IDENTITY ELEMENT WHEN DETERMINING
C     THE PUMPED I-V CURVE CURRENT VALUE
C
      CURPMP = CURPMP + REAL(CTEMP(1)) / RID
C
      INTER = -IBASE
      CALL FFT(INTER,CTEMP,FACTOR)
C
      DO 30 I = 1,NOPNTS
30    VOLT(I) = REAL(CTEMP(I))
C
      RETURN
```

FILE: SISCAP   FORTRAN A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

```
      END
      SUBROUTINE SISNL(VOLT,CUR,NOPNTS,IBASE1,CHI,NOCHI,RN,FACTOR,FREQ,
     1RID)
C
      REAL VOLT(NOPNTS), CHI(NOCHI), CUR(NOPNTS)
      COMPLEX UTERM(1Ø24), FACTOR(NOPNTS), CVOL(128), CSUM, CTEMP
      COMPLEX CCHI(1Ø24)
      DATA PI,E,HBAR/3.1415926,1.6Ø2E-19,1.Ø55E-34/
C
C     THIS IS THE SIS NONLINEAR NETWORK CALCULATION SUBROUTINE
C     SUBROUTINE PART OF THE MULTIPLE REFLECTION TECHNIQUE
C
      ITST1 = 1Ø
      IF(NOCHI.EQ.1Ø24) ITST1 = 11
      ITST2 = -ITST1
C
      IF(NOCHI.GT.1Ø24) GO TO 1ØØØ
      IF(NOCHI.NE.1Ø24 .AND. NOCHI.NE.512) GO TO 1ØØ2
C
C     SET UP A FEW PARAMETERS
C
      OMEGA = 2. * PI * FREQ
      NODIV2 = NOPNTS/2
      INTER = -IBASE1
      CONST = -1. * E / HBAR / OMEGA
      HSTEP = 1. / FREQ / FLOAT(NOPNTS)
C
C     CALCULATE THE DC SEPARATELY
C
      DO 1ØØ I = 1,NOPNTS
1ØØ   CVOL(I) = VOLT(I)
      CALL FFT(IBASE1,CVOL,FACTOR)
      RAMP = REAL(CVOL(1)) * (-1.) * E / HBAR
C
C     INTEGRATE THE NON-DC COMPONENTS
C
      DO 2ØØ I = 2,NODIV2
      ICORR = NOPNTS - I + 2
      CVOL(I) = CVOL(I) / CMPLX(Ø.,1.) / FLOAT(I-1) * CONST
      CVOL(ICORR) = CONJG(CVOL(I))
2ØØ   CONTINUE
      CVOL(1) = CMPLX(Ø.,Ø.)
      CVOL(NODIV2+1) = CMPLX(Ø.,Ø.)
C
      CALL FFT(INTER,CVOL,FACTOR)
C
C      NTIME = NOCHI / NOPNTS + 1
C
C     CALCULATE U
C
      NTIME = 2 * NOCHI / NOPNTS
      DO 3ØØ J = 1,NTIME
      DO 3ØØ I = 1,NOPNTS
      IPTR = NOPNTS * (J-1) + I
      TEMP = REAL(CVOL(I)) + FLOAT(IPTR-1) * HSTEP * RAMP
```

```
FILE: SISCAP   FORTRAN  A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

      IF(ABS(TEMP).LE.2.*PI) GO TO 25Ø                                  SISØ496Ø
      IREVS = TEMP/2./PI                                                SISØ497Ø
      TEMP = TEMP - FLOAT(IREVS) * 2. * PI                              SISØ498Ø
25Ø   CTEMP = CMPLX(Ø.,TEMP)                                            SISØ499Ø
      UTERM(IPTP) = CEXP(CTEMP)                                         SISØ5ØØØ
3ØØ   CONTINUE                                                          SISØ5Ø1Ø
C                                                                       SISØ5Ø2Ø
      DO 5ØØØ IMF = 1,NOCHI                                             SISØ5Ø3Ø
      CCHI(IMF) = CHI(IMF)                                              SISØ5Ø4Ø
      CCHI(IMF+NOCHI) = Ø.Ø                                             SISØ5Ø5Ø
5ØØØ  CONTINUE                                                          SISØ5Ø6Ø
C                                                                       SISØ5Ø7Ø
C     DO THE CONVOLUTION CALCULATIONS IN THE FREQUENCY DOMAIN           SISØ5Ø8Ø
C                                                                       SISØ5Ø9Ø
      CALL FFT(ITST1,CCHI,FACTOR)                                       SISØ51ØØ
      CALL FFT(ITST1,UTERM,FACTOR)                                      SISØ511Ø
C                                                                       SISØ512Ø
      NOCHM2 = 2 * NOCHI                                                SISØ513Ø
      DO 6ØØØ IMF = 1,NOCHM2                                            SISØ514Ø
      CCHI(IMF) = CCHI(IMF) * UTERM(IMF) * FLOAT(NOCHM2)                SISØ515Ø
6ØØØ  CONTINUE                                                          SISØ516Ø
C                                                                       SISØ517Ø
      CALL FFT(ITST2,CCHI,FACTOR)                                       SISØ518Ø
      CALL FFT(ITST2,UTERM,FACTOR)                                      SISØ519Ø
C                                                                       SISØ52ØØ
C     PERFORM THE FINAL CALCULATIONS AND CORRECT FOR THE                SISØ521Ø
C     IDENTITY ELEMENT VALUE                                            SISØ522Ø
C                                                                       SISØ523Ø
      DO 6ØØ I = 1,NOPNTS                                               SISØ524Ø
      N = 2*NOCHI - NOPNTS + I                                          SISØ525Ø
C      CSUM = CMPLX(Ø.,Ø.)                                              SISØ526Ø
C      N = NOCHI + 1                                                    SISØ527Ø
C      NOMIN1 = NOCHI - 1                                               SISØ528Ø
C      NOMIN3 = NOCHI - 3                                               SISØ529Ø
C      DO 5ØØ K = 1,NOMIN3,2                                            SISØ53ØØ
C      CSUM = (CHI(K)*UTERM(N-K+1)+4.Ø*CHI(K+1)*UTERM(N-K)+             SISØ531Ø
C     1CHI(K+2)*UTERM(N-K-1))/3.Ø + CSUM                                SISØ532Ø
C5ØØ   CONTINUE                                                         SISØ533Ø
C      CSUM = CSUM+(CHI(NOMIN1)*UTERM(N-NOCHI+2)+4.Ø*CHI(NOCHI)         SISØ534Ø
C     1*UTERM(N-NOCHI+1))/3 Ø                                           SISØ535Ø
      CSUM = CCHI(N)                                                    SISØ536Ø
      CSUM = CSUM * CONJG(UTERM(N)) * HSTEP                             SISØ537Ø
      CUR(I) = VOLT(I) / RN + AIMAG(CSUM) - VOLT(I) / RID               SISØ538Ø
6ØØ   CONTINUE                                                          SISØ539Ø
C                                                                       SISØ54ØØ
      RETURN                                                            SISØ541Ø
C                                                                       SISØ542Ø
1ØØØ  WRITE(6,1ØØ1) NOCHI                                               SISØ543Ø
1ØØ1  FORMAT(1X,'NOCHI = ',I5,' AND IS TOO BIG FOR THE AVAIL MEMORY')   SISØ544Ø
      STOP                                                              SISØ545Ø
C                                                                       SISØ546Ø
1ØØ2  WRITE(6,1ØØ3) NOCHI                                               SISØ547Ø
1ØØ3  FORMAT(1X,'NOCHI = ',I5,' AND IS AN INCORRECT VALUE')             SISØ548Ø
      STOP                                                              SISØ549Ø
      END                                                               SISØ55ØØ
```

```
FILE: SISCAP   FORTRAN  A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

      SUBROUTINE COMPAR(LATEST,OLD,NOPNTS,FREITR)                       SISØ551Ø
      REAL LATEST(NOPNTS), OLD(NOPNTS)                                  SISØ552Ø
      LOGICAL FREITR                                                    SISØ553Ø
      COMMON /PERIPH/IN,IOUT                                            SISØ554Ø
      COMMON /ERRALL/  TOL, EPS                                         SISØ555Ø
C                                                                       SISØ556Ø
C  AUTHOR: R.G. HICKS, UNIV. OF QLD., APRIL 1977                        SISØ557Ø
C                                                                       SISØ558Ø
C     AIM: THIS SUBROUTINE COMPARES TWO VECTORS 'LATEST'                SISØ559Ø
C          AND 'OLD' AND PROVIDES A LOGIC ANSWER 'FREITR'               SISØ56ØØ
C          AS TO WHETHER THE TWO VECTORS ARE SUFFICIENTLY EQUAL.        SISØ561Ø
C                                                                       SISØ562Ø
      FREITR = .FALSE.                                                  SISØ563Ø
      ERR = Ø.Ø                                                         SISØ564Ø
C                                                                       SISØ565Ø
      TOP = Ø.                                                          SISØ566Ø
      DO 5 I = 1,NOPNTS                                                 SISØ567Ø
      IF(ABS(LATEST(I)).GT.TOP) TOP = ABS(LATEST(I))                    SISØ568Ø
5     CONTINUE                                                          SISØ569Ø
C                                                                       SISØ57ØØ
      DO 1Ø  I=1,NOPNTS                                                 SISØ571Ø
      IF(ABS(LATEST(I)).LE.EPS*TOP) GO TO 1Ø                            SISØ572Ø
      TEST = (LATEST(I) - OLD(I))/LATEST(I)                             SISØ573Ø
      IF(ABS(TEST).GT.TOL)  FREITR = .TRUE.                             SISØ574Ø
      IF(ABS(TEST) .LE. ERR) GO TO 1Ø                                   SISØ575Ø
      ERR = ABS(TEST)                                                   SISØ576Ø
      MEM = I                                                           SISØ577Ø
1Ø    CONTINUE                                                          SISØ578Ø
C                                                                       SISØ579Ø
      WRITE(IOUT,2Ø) ERR, MEM                                           SISØ58ØØ
2Ø    FORMAT(8X,'ERR = ',E15.8,'POSN NO = ',I5)                         SISØ581Ø
C                                                                       SISØ582Ø
      RETURN                                                            SISØ583Ø
      END                                                               SISØ584Ø
      SUBROUTINE TOPS(FREQ,FIF,IBASE2,NOIV,IBASE1,NOPNTS,MMAX,CAP,      SISØ585Ø
     1RN,TEMP,NMAX,VOLT,YØ,Y1,Y1M,ZLO,IVDAT,J,FACTOR,DELTAV,ISIG,ZSSB)  SISØ586Ø
      REAL IVDAT(NOIV), VOLT(NOPNTS)                                    SISØ587Ø
      COMPLEX YSIG(31,31), NOISE(31,31), ZSIG(31,31), ZOUT              SISØ588Ø
      COMPLEX B(31), STORE(31,31)                                       SISØ589Ø
      COMPLEX FACTOR(NOIV), YØ, Y1,Y1M, ZØ, ZSSB(MMAX)                  SISØ59ØØ
      COMPLEX J(NOIV), W(128)                                           SISØ591Ø
      COMPLEX YSI2(3,3), NOI2(3,3), ZSI2(3,3)                           SISØ592Ø
      COMPLEX YCAP, ZLO                                                 SISØ593Ø
      COMMON /PERIPH/ IN, IOUT, IPLOT                                   SISØ594Ø
      COMMON /RBOUT/ VDC, VLO, VJDC, VJLO, CURDC, CURPMP, RLOSS, RTEMP  SISØ595Ø
C                                                                       SISØ596Ø
C     THIS IS THE MAIN SMALL SIGNAL CONVERSION AND NOISE                SISØ597Ø
C     CONTROLLING CALCULATION ROUTINE                                   SISØ598Ø
C                                                                       SISØ599Ø
      WRITE(6,7Ø9) VJDC, VJLO                                           SISØ6ØØØ
7Ø9   FORMAT(1X,'DC VOLTAGE AT THE JUNCTION',8X,'= ',E15.8,' VOLTS'/    SISØ6Ø1Ø
     11X,'LO VOLTAGE (PEAK) AT THE JUNCTION = ',E15.8,' VOLTS')         SISØ6Ø2Ø
C                                                                       SISØ6Ø3Ø
      WRITE(6,1)                                                        SISØ6Ø4Ø
1     FORMAT(/1X,'SMALL SIGNAL AND NOISE ANALYSIS SECTION'/)            SISØ6Ø5Ø
```

```
C
      IF(ISIG.EQ.1) GO TO 55ØØ
C
C     FOR THE GENERAL ANALYSIS METHOD
C
      DO 1Ø I = 1,MMAX
      B(I) = CMPLX(Ø.,Ø.)
      DO 1Ø II = 1,MMAX
      ZSIG(I,II) = CMPLX(Ø.,Ø.)
      STORE(I,II) = CMPLX(Ø.,Ø.)
      NOISE(I,II) = CMPLX(Ø.,Ø.)
      YSIG(I,II) = CMPLX(Ø.,Ø.)
1Ø    CONTINUE
C
      WRITE(6,5Ø4)
5Ø4   FORMAT(/1X,'DETERMINING SMALL SIGNAL MIXER PROPERTIES USING',
     1' THE GENERAL ANALYSIS')
C
      CALL SIGNL(IVDAT,NOIV,FREQ,IBASE1,IBASE2,VOLT,NOPNTS,YSIG,MMAX,
     1FIF,DELTAV,J,W,FACTOR,NMAX,RN)
C
      CALL SIGLIN(FREQ,YSIG,MMAX,FIF,CAP,ZSSB)
C
      CALL INVRT(MMAX,YSIG,ZSIG,B,STORE)
C
      CALL LOSCAL(MMAX,YSIG,ZSIG,YØ,Y1,Y1M)
C
      CALL SISNOI(W,NMAX,MMAX,NOIV,FREQ,FIF,VJDC,
     1J,DELTAV,TEMP,NOISE,NOPNTS,RN)
      CALL PESNOI(TEMP,NOISE,MMAX,FREQ,FIF,CAP,ZSSB)
      CALL PROCES(ZSIG,NOISE,MMAX,Y1,Y1M)
C
      GO TO 5ØØØ
C
C     USING THE APPROXIMATE THREE-PORT MODEL
C
55ØØ  DO 15 I = 1,3
      DO 15 II = 1,3
      NOI2(I,II) = CMPLX(Ø.,Ø.)
      ZSI2(I,II) = CMPLX(Ø.,Ø.)
      YSI2(I,II) = CMPLX(Ø.,Ø.)
15    CONTINUE
      DO 17 I = 1,MMAX
      B(I) = CMPLX(Ø.,Ø.)
      DO 17 II = 1,MMAX
      STORE(I,II) = CMPLX(Ø.,Ø.)
17    CONTINUE
C
      WRITE(6,5Ø5)
5Ø5   FORMAT(/1X,'THREE PORT SMALL SIGNAL ANALYSIS USING FORMULAE')
C
      CALL THEORY(VJLO,VJDC,NMAX,IVDAT,NOIV,J,FREQ,YSI2,NOI2,TEMP,
     1DELTAV,RN,YØ,Y1,Y1M,ZLO)
C
4999  WRITE(6,5Ø6)
```

FILE: SISCAP   FORTRAN A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

```
5Ø6    FORMAT(/1X,'THREE PORT ANALYSIS USING MATRIX TECHNIQUES')
C
       CALL SIGLIN(FREQ,YSI2,3,FIF,CAP,ZSSB)
       CALL INVRT(3,YSI2,ZSI2,B,STORE)
       CALL LOSCAL(3,YSI2,ZSI2,YØ,Y1,Y1M)
       CALL RESNOI(TEMP,NOI2,3,FREQ,FIF,CAP,ZSSB)
       CALL PROCES(ZSI2,NOI2,3,Y1,Y1M)
C
5ØØØ   RETURN
C
       END
       SUBROUTINE RESNOI(TEMP,NOISE,MMAX,FREQ,FIF,CAP,ZSSB)
       REAL K
       COMPLEX NOISE(MMAX,MMAX), YPAR, ZSSB(MMAX)
       DATA HBAR/1.Ø55E-34/
       DATA PI/3.1415926/
       DATA K/1.38Ø622E-23/
C
C      DETERMINE THE THERMAL NOISE COMPONENT EMANATING FROM
C      THE RESISTANCES USING THE QUANTUM CORRECTED FORMULA
C      NO NOISE FROM THE EMBEDDING IMPEDANCES AT THE SIGNAL
C      AND IMAGE IS CONSIDERED IN THE DEFINITION
C
       DO 1ØØ I = 1,MMAX
       M = (MMAX/2) - I + 1
       IPTR = IABS(M) * 2
       IF(M.LE.Ø) IPTR = IPTR + 1
       IF(CABS(ZSSB(IPTR)).LT.1.E-1Ø) GO TO 5ØØ
       YPAR = 1. / ZSSB(IPTR)
       GO TO 6ØØ
5ØØ    YPAR = CMPLX(Ø.Ø,1.E1Ø)
6ØØ    IF(IABS(M).EQ.1) YPAR = CMPLX(Ø.Ø,Ø.Ø)
       IF(M.EQ.Ø) YPAR = CMPLX(Ø.Ø,Ø.Ø)
       OMEGAM = (FLOAT(M) * FREQ + FIF) * 2.Ø * PI
       VAL = HBAR * OMEGAM / K /TEMP
       NOISE(I,I) = NOISE(I,I) + 4.Ø*HBAR*OMEGAM*REAL(YPAR)
      1 / (EXP(VAL) - 1.Ø)
1ØØ    CONTINUE
       RETURN
       END
       SUBROUTINE PROCES(ZSIG,NOISE,MMAX,Y1,Y1M)
       REAL K
       COMPLEX STORE(31). SUM
       COMPLEX Y1, Y1M, NOISE(MMAX,MMAX), ZSIG(MMAX,MMAX)
       COMPLEX ZIN
       INTEGER SGPORT
       COMMON /RBOUT/ VDC, VLO, VJDC, VJLO, CURDC, CURPMP, RLOSS, TEMP1
       DATA K/1 38Ø622E-23/, TOL/1.E-4/
C
C      DETERMINE THE NOISE TEMPEPATURE USING THE IMPEDANCE
C      MATRIX AND THE NOISE CONTRIBUTIONS MATRIX
C
       IFPORT = MMAX/2 + 1
       SGPORT = MMAX/2
C
```

```
      DO 2Ø I = 1,MMAX
      SUM = CMPLX(Ø.,Ø.)
      DO 1Ø J = 1,MMAX
      SUM = SUM + NOISE(I,J) * CONJG(ZSIG(IFPORT,J))
1Ø    CONTINUE
      STORE(I) = SUM
2Ø    CONTINUE
C
      SUM = CMPLX(Ø.,Ø.)
C
      DO 3Ø I = 1,MMAX
      SUM = SUM + ZSIG(IFPORT,I) * STORE(I)
3Ø    CONTINUE
C
      VREAL = REAL(SUM)
      VIMAG = AIMAG(SUM)
      IF(ABS(VIMAG).GT.TOL*ABS(VREAL)) GO TO 1ØØ
C
      VAL1 = CABS(ZSIG(IFPORT,SGPORT))
      VAL2 = CABS(ZSIG(IFPORT,IFPORT+1))
C
C      POWER = REAL(SUM) / 5Ø.Ø
C      WRITE(6,5Ø) POWER
C5Ø    FORMAT(8X,'IF OUTPUT POWER INTO 5Ø OHM LOAD = ',E15.8)
C
      TEMP1 = REAL(SUM) / (4.*K*REAL(YI)*VAL1**2)
      TEMP2 = REAL(SUM) / (4.*K*REAL(YIM)*VAL2**2)
      WRITE(6,4Ø) TEMP1, TEMP2
4Ø    FORMAT(/8X,'SSB TEMP OF MIXER (USB) = ',E15.8,' DEG K'/
     18X,'SSB TEMP OF MIXER (LSB) = ',E15.8,' DEG K')
C
      RETURN
C
1ØØ   WRITE(6,11Ø) VREAL, VIMAG
11Ø   FORMAT(1X,'PROBLEM IN TEMPERATURE CALCULATIONS:'/
     11X,'TEMP = ',E15.8,' +J ',E15.8)
      RETURN
      END
      SUBROUTINE SIGNL(IVDAT,NOIV,FREQ,IBASE1,IBASE2,VOLT,NOPNTS,YSIG,
     1MMAX,FIF,DELTAV,J,W,FACTOR,NMAX,RN)
      REAL IVDAT(NOIV), VOLT(NOPNTS)
      COMPLEX CVOL(128), J(NOIV), W(NOPNTS), YSIG(MMAX,MMAX)
      COMPLEX CVAL1, CVAL2, CVAL3, CVAL4, CSUM
      COMPLEX FACTOR(NOIV)
      COMMON /RBOUT/ VDC, VLO, VJDC, VJLO, CURDC, CURPMP, RLOSS, RTEMP
      DATA PI,E,HBAR/3.1415926,1.6Ø2E-19,1.Ø55E-34/
C
C     DETERMINE THE ADMITTANCE CONVERSION COMPONENTS DUE TO
C     THE SIS TUNNEL JUNCTION AS PER TUCKERS THEORY
C
      OMEGA = 2. * PI * FREQ
      NODIV2 = NOPNTS/2
      INTER1 = -IBASE1
      INTER2 = -IBASE2
      CONST = -1. * E / HBAR / OMEGA
      HSTEP = 1. / FREQ / FLOAT(NOPNTS)
      WIF = 2. * PI * FIF
C
      DO 1ØØ I = 1,NOPNTS
1ØØ   CVOL(I) = VOLT(I)
      CALL FFT(IBASE1,CVOL,FACTOR)
C
      VBIAS = REAL(CVOL(1))
C
      DO 2ØØ I = 2,NODIV2
      ICORR = NOPNTS - I + 2
      CVOL(I) = CVOL(I) / CMPLX(Ø.,1.) / FLOAT(I-1) * CONST
      CVOL(ICORR) = CONJG(CVOL(I))
2ØØ   CONTINUE
C
      CVOL(1) = CMPLX(Ø.,Ø.)
      CVOL(NODIV2+1) = CMPLX(Ø.,Ø.)
C
      CALL FFT(INTER1,CVOL,FACTOR)
C
      DO 3ØØ I = 1,NOPNTS
      TEMP = REAL(CVOL(I))
      CVOL(I) = CMPLX(Ø.,TEMP)
      W(I) = CEXP(CVOL(I))
3ØØ   CONTINUE
C
      CALL FFT(IBASE1,W,FACTOR)
C
      VNORM = E * VBIAS / HBAR
C
      DO 1ØØØ M1 = 1,MMAX
      DO 1ØØØ M2 = 1,MMAX
      M = MMAX/2 - M1 + 1
      MDASH = MMAX/2 - M2 + 1
      IDIF = M - MDASH
      WMDASH = FLOAT(MDASH) * OMEGA + WIF
      CSUM = CMPLX(Ø.,Ø.)
      DO 5ØØ N1 = 1,NMAX
      NDASH = NMAX/2 - N1 + 1
      N = NDASH - IDIF
      IF(IABS(N).GT.NMAX/2) GO TO 5ØØ
      ARG1 = FLOAT(N) * OMEGA + VNORM
      ARG2 = FLOAT(N) * OMEGA - WMDASH + VNORM
      ARG3 = FLOAT(NDASH) * OMEGA + WMDASH + VNORM
      ARG4 = FLOAT(NDASH) * OMEGA + VNORM
      CALL EVALJ(ARG1,CVAL1,J,NOIV,DELTAV,RN)
      CALL EVALJ(ARG2,CVAL2,J,NOIV,DELTAV,RN)
      CALL EVALJ(ARG3,CVAL3,J,NOIV,DELTAV,RN)
      CALL EVALJ(ARG4,CVAL4.J,NOIV,DELTAV,RN)
      IF(ABS(WMDASH).GT 1.ØE9*2.Ø*PI) GO TO 4356
      CALL DERIJ(ARG1,CVAL1,J,NOIV,DELTAV,RN)
      CALL DERIJ(ARG4,CVAL4,J,NOIV,DELTAV,RN)
      CVAL4 = CMPLX(-1.Ø,Ø.Ø) * CVAL4
      CVAL2 = CMPLX(Ø.Ø,Ø.Ø)
      CVAL3 = CMPLX(Ø.Ø,Ø.Ø)
```

```
4356  NPTR1 = N + 1                                                     SISØ826Ø
      IF(IABS(N).GE.NOPNTS/2) GO TO 5ØØ                                 SISØ827Ø
      IF(NPTR1.LE.Ø) NPTR1 = NPTR1 + NOPNTS                             SISØ828Ø
      NPTR2 = NDASH + 1                                                 SISØ829Ø
      IF(IABS(NDASH).GE.NOPNTS/2) GO TO 5ØØ                             SISØ83ØØ
      IF(NPTR2.LE.Ø) NPTR2 = NPTR2 + NOPNTS                             SISØ831Ø
      CSUM = W(NPTR1) * CONJG(W(NPTR2)) * (CVAL1-CVAL2-CONJG(CVAL3)     SISØ832Ø
     1 + CONJG(CVAL4)) + CSUM                                           SISØ833Ø
5ØØ   CONTINUE                                                          SISØ834Ø
      YSIG(M1,M2) = CMPLX(Ø.,-1.)*E/2./HBAR/WMDASH*CSUM                 SISØ835Ø
      IF(ABS(WMDASH).GT.1.ØE9*2.Ø*PI) GO TO 1ØØØ                        SISØ836Ø
      YSIG(M1,M2) = CMPLX(Ø.Ø,-1.Ø) / 2.Ø * CSUM                        SISØ837Ø
C1963  WRITE(6,2831) M1,M2, YSIG(M1,M2)                                 SISØ838Ø
C2831  FORMAT(1X,'M1=',I5,'M2=',I5,'YSIG=',2E15.8)                      SISØ839Ø
1ØØØ  CONTINUE                                                          SISØ84ØØ
      RETURN                                                            SISØ841Ø
      END                                                               SISØ842Ø
      SUBROUTINE GETJ(IVDAT,NOIV,FREQ,IBASE2,J,FACTOR,RN,DELTAV)        SISØ843Ø
      COMPLEX CTEMP(2Ø48), FACTOR(NOIV), J(NOIV)                        SISØ844Ø
      REAL IVDAT(NOIV)                                                  SISØ845Ø
C                                                                       SISØ846Ø
C     OBTAIN THE VECTOR J BY PERFORMING A KRONERS KRONIG                SISØ847Ø
C     TRANSFORM ON THE REDUCED I-V CURVE AS PER THE REPORT              SISØ848Ø
C                                                                       SISØ849Ø
      IBASE3 = IBASE2 + 1                                               SISØ85ØØ
      NOUSE = 2 * NOIV                                                  SISØ851Ø
      INTER3 = -IBASE3                                                  SISØ852Ø
C                                                                       SISØ853Ø
      DO 1ØØ  I = 1,NOIV                                                SISØ854Ø
      CTEMP(I) = IVDAT(I)                                               SISØ855Ø
      IF(I.EQ.1) GO TO 1ØØ                                              SISØ856Ø
      ICORR = NOUSE - I + 2                                             SISØ857Ø
      CTEMP(ICORR) = -IVDAT(I)                                          SISØ858Ø
1ØØ   CONTINUE                                                          SISØ859Ø
      CTEMP(NOIV+1) = CMPLX(Ø.,Ø.)                                      SISØ86ØØ
C                                                                       SISØ861Ø
      CALL FFT(IBASE3,CTEMP,FACTOR)                                     SISØ862Ø
C                                                                       SISØ863Ø
      DO 2ØØ I = 2,NOIV                                                 SISØ864Ø
      CTEMP(I) = CMPLX(Ø.,1.) * CTEMP(I)                                SISØ865Ø
      ICORR = NOUSE - I + 2                                             SISØ866Ø
      CTEMP(ICORR) = CMPLX(Ø.,-1.) * CTEMP(ICORR)                       SISØ867Ø
2ØØ   CONTINUE                                                          SISØ868Ø
C                                                                       SISØ869Ø
      CTEMP(1) = CMPLX(Ø.,Ø.)                                           SISØ87ØØ
      CTEMP(NOIV+1) = CMPLX(Ø.,Ø.)                                      SISØ871Ø
C                                                                       SISØ872Ø
      CALL FFT(INTER3,CTEMP,FACTOR)                                     SISØ873Ø
C                                                                       SISØ874Ø
      DIFF = REAL(CTEMP(1))                                             SISØ875Ø
C                                                                       SISØ876Ø
      DO 3ØØ I = 1,NOIV                                                 SISØ877Ø
      TEMP = REAL(CTEMP(I)) - DIFF                                      SISØ878Ø
      HOLD = IVDAT(I)                                                   SISØ879Ø
      J(I) = CMPLX(TEMP,HOLD)                                           SISØ88ØØ
```

```
3ØØ   CONTINUE
C
      RETURN
C
      END
      SUBROUTINE SIGLIN(FREQ,YSIG,MMAX,FIF,CAP,ZSSB)
      COMPLEX YSIG(MMAX,MMAX), ZSSB(MMAX), CVAL
C
C     OBTAIN THE CONTRIBUTIONS TO THE SIGNAL CONVERSION MATRIX
C     FROM THE EMBEDDING IMPEDANCES
C
      DO 2ØØØ M1 = 1,MMAX
      M = MMAX/2 - M1 + 1
C
      IPTR = IABS(M) * 2
      IF(M.LE.Ø) IPTR = IPTR + 1
      IF(CABS(ZSSB(IPTR)).LT.1.E-1Ø) GO TO 5ØØ
      CVAL = 1. / ZSSB(IPTR)
      GO TO 6ØØ
5ØØ   CVAL = CMPLX(1.E1Ø,1.E1Ø)
6ØØ   YSIG(M1,M1) = YSIG(M1,M1) + CVAL
2ØØØ  CONTINUE
      RETURN
      END
      SUBROUTINE LOSCAL(MMAX,YSIG,ZSIG,YØ,Y1,Y1M)
C
      INTEGER SGPORT
      REAL LOSS, LOSSL
      REAL MSMCHI, MSMCHO
      COMPLEX Z11, ZIN, CTMP1, ZSOURC, ZOUT
      COMPLEX YØ,Y1,YSIG(MMAX,MMAX),ZSIG(MMAX,MMAX),ZØ1, ZØØ, RHO
      COMPLEX Y1M, ZØM1
      COMMON /REOUT/ VDC, VLO, VJDC, VJLO, CURDC, CURPMP, LOSS, RTEMP
C
C     CALCULATE THE CONVERSION LOSS FROM THE SMALL SIGNAL
C     IMPEDANCE CONVERSION MATRIX
C
      RSOURC = 1.Ø/REAL(Y1)
      RØ = 1.Ø / REAL(YØ)
C
      IFPORT = MMAX/2 + 1
      SGPORT = MMAX/2
      IMPORT = MMAX/2 + 2
      ZØ1 = ZSIG(IFPORT,SGPORT)
      ZØØ = ZSIG(IFPORT,IFPORT)
      ZØM1 = ZSIG(IFPORT,IMPORT)
      Z11 = ZSIG(SGPORT,SGPORT)
C
      ZOUT = 1.Ø / (1.Ø/ZØØ - 1.Ø/RØ)
      RHO = (ZOUT - RØ) / (ZOUT + RØ)
      VSWR = (1.Ø + CABS(RHO)) / (1.Ø - CABS(RHO))
      WRITE(6,555) ZOUT, RØ, VSWR
555   FORMAT(/8X,'IF OUTPUT IMPEDANCE = ',E15.8,' +J ',E15.8/
     18X,'IF VSWR REFERRED TO ',F5.2,' OHMS = ',E15.8)
C
```

```
FILE: SISCAP   FORTRAN  A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

      ZIN = 1.Ø / (1.Ø/Z11 - 1.Ø/RSOURC)                                SISØ936Ø
      CTMP1 = RSOURC + ZIN                                              SISØ937Ø
      MSMCHI = 4.Ø*RSOURC*REAL(ZIN)/(CABS(CTMP1)**2)                    SISØ938Ø
      DB1 = ABS(MSMCHI)                                                 SISØ939Ø
      DB1 = 1Ø.Ø * ALOG1Ø(DB1)                                          SISØ94ØØ
      IF(MSMCHI.LT.Ø.Ø) DB1 = Ø.Ø                                       SISØ941Ø
C                                                                       SISØ942Ø
      CTMP1 = RØ + ZOUT                                                 SISØ943Ø
      MSMCHO = 4.Ø*RØ*REAL(ZOUT)/(CABS(CTMP1)**2)                       SISØ944Ø
      DB2 = ABS(MSMCHO)                                                 SISØ945Ø
      DB2 = 1Ø.Ø*ALOG1Ø(DB2)                                            SISØ946Ø
      IF(MSMCHO.LT.Ø.Ø) DB2 = Ø.Ø                                       SISØ947Ø
C                                                                       SISØ948Ø
      WRITE(6,4ØØØ) MSMCHO, DB2, ZIN, MSMCHI, DB1, RSOURC               SISØ949Ø
4ØØØ  FORMAT(8X,'OUTPUT MISMATCH GAIN = ',E15.8,' (',E15.8,'DB)'//      SISØ95ØØ
     18X,'INPUT SIGNAL IMPEDANCE = ',E15.8,' +J ',E15.8,'OHMS'/         SISØ951Ø
     28X,'INPUT MISMATCH GAIN = ',E15.8,' (',E15.8,'DB)'/               SISØ952Ø
     38X,'INPUT GAIN CALCULATION REFERRED TO = ',E15.8,' OHMS')         SISØ953Ø
C                                                                       SISØ954Ø
      LOSS = Ø.25/CABS(ZØ1)**2/REAL(Y1)/REAL(YØ)                        SISØ955Ø
      LOSS = -1Ø. * ALOG1Ø(LOSS)                                        SISØ956Ø
      LOSSL = Ø.25/CABS(ZØM1)**2/REAL(Y1M)/REAL(YØ)                     SISØ957Ø
      LOSSL = -1Ø.Ø * ALOG1Ø(LOSSL)                                     SISØ958Ø
      WRITE(6,3ØØØ) LOSS, LOSSL                                         SISØ959Ø
3ØØØ  FORMAT(/8X,'CONVERSION GAIN (USB) EQUALS ',E15.8,' DB'/           SISØ96ØØ
     18X,'CONVERSION GAIN (LSB) EQUALS ',E15.8,' DB')                   SISØ961Ø
      RETURN                                                            SISØ962Ø
      END                                                               SISØ963Ø
      SUBROUTINE EVALJ(DUM,CVAL,J,NOIV,DELTAV,RN)                       SISØ964Ø
      COMPLEX CVAL, J(NOIV), TERP, TERP2, TERP3, TERP5                  SISØ965Ø
      LOGICAL FLAG                                                      SISØ966Ø
      DATA E, HBAR/1.6Ø2E-19,1.Ø55E-34/                                 SISØ967Ø
C                                                                       SISØ968Ø
C     EVALUATE THE REQUIRED VALUE OF J BY INTERPOLATING                 SISØ969Ø
C     THROUGH THE ARRAY OF CALCULATED J VALUES.                         SISØ97ØØ
C     FOR NEGATIVE ARGUMENTS, THE CONJUGATE VALUE OF J                  SISØ971Ø
C     MUST BE USED.                                                     SISØ972Ø
C                                                                       SISØ973Ø
      FLAG = .FALSE.                                                    SISØ974Ø
      ARG = DUM                                                         SISØ975Ø
      IF(ARG) 1ØØ,4ØØ,3ØØ                                               SISØ976Ø
1ØØ   FLAG = .TRUE.                                                     SISØ977Ø
      ARG = -DUM                                                        SISØ978Ø
3ØØ   PTR = ARG / DELTAV / E * HBAR                                     SISØ979Ø
      IPTR = PTR + 1.                                                   SISØ98ØØ
      IF(IPTR.GE.NOIV-1) GO TO 2ØØ                                      SISØ981Ø
      RATIO = AMOD(PTR,1.)                                              SISØ982Ø
      CVAL = TERP5(J,NOIV,IPTR,RATIO,.FALSE.)                           SISØ983Ø
      TEMP = PTR * DELTAV / RN                                          SISØ984Ø
      CVAL = CVAL + CMPLX(Ø.,TEMP)                                      SISØ985Ø
      IF(FLAG)CVAL = CONJG(CVAL)                                        SISØ986Ø
      RETURN                                                            SISØ987Ø
2ØØ   TEMP = PTR * DELTAV / RN                                          SISØ988Ø
      CVAL = J(NOIV) + CMPLX(Ø.,TEMP)                                   SISØ989Ø
      IF(FLAG) CVAL = CONJG(CVAL)                                       SISØ99ØØ
      RETURN
4ØØ   CVAL = CMPLX(Ø.,Ø.)
      RETURN
      END
      SUBROUTINE SISNOI(W,NMAX,MMAX,NOIV,FREQ,FIF,VBIAS,J,DELTAV,
     1TEMP,NOISE,NOPNTS,RN)
      COMPLEX W(NOPNTS), NOISE(MMAX,MMAX), J(NOIV), CVAL3, CVAL4
      DATA PI,E,HBAR/3.1415926,1.6Ø2E-19,1.Ø55E-34/
      DATA BOLTZ/1.38Ø622E-23/
C
C     EVALUATE THE NOISE COMPONENT EMANATING FROM THE
C     SHOT NOISE IN THE TUNNEL JUNCTION AS PER TUCKERS
C     THEORY.
C
      BETA = 1. / BOLTZ / TEMP
      OMEGA = 2. * PI * FREQ
      WIF = 2. * PI * FIF
      CONST = E / HBAR
C
      DO 1ØØØ M1 = 1,MMAX
      DO 1ØØØ M2 = 1,MMAX
      M = MMAX/2 - M1 + 1
      MDASH = MMAX/2 - M2 + 1
      WMDASH = FLOAT(MDASH) * OMEGA + WIF
      IADJ = M - MDASH
      DO 5ØØ N1 = 1,NMAX
      N = NMAX/2 - N1 + 1
      NDASH = N + IADJ
      ARG1 = BETA*(E*VBIAS+FLOAT(NDASH)*HBAR*OMEGA+HBAR*WMDASH)/2.
      ARG2 = BETA*(E*VBIAS+FLOAT(N)*HBAR*OMEGA-HBAR*WMDASH)/2.
      ARG3 = VBIAS+FLOAT(NDASH)*HBAR*OMEGA/E + HBAR*WMDASH/E
      ARG4 = VBIAS + FLOAT(N)*HBAR*OMEGA/E - HBAR*WMDASH/E
      ARG3 = CONST * ARG3
      ARG4 = CONST * ARG4
      CALL EVALJ(ARG3,CVAL3,J,NOIV,DELTAV,RN)
      CALL EVALJ(ARG4,CVAL4,J,NOIV,DELTAV,RN)
      VAL3 = AIMAG(CVAL3)
      VAL4 = AIMAG(CVAL4)
      VAL1 = COTH(ARG1)
      VAL2 = COTH(ARG2)
      NPTR1 = N + 1
      IF(NPTR1.LE.Ø) NPTR1 = NPTR1 + NOPNTS
      NPTR2 = NDASH + 1
      IF(NPTR2.LE.Ø) NPTR2 = NPTR2 + NOPNTS
      NOISE(M1,M2) = NOISE(M1,M2) + E*W(NPTR1)*CONJG(W(NPTR2))*(VAL1
     1 * VAL3 + VAL2*VAL4)
5ØØ   CONTINUE
1ØØØ  CONTINUE
      RETURN
      END
      SUBROUTINE INVRT(NOSIG,YSIG,ZSIG,B,STORE)
C
      COMPLEX YSIG(NOSIG,NOSIG), ZSIG(NOSIG), B(NOSIG)
      COMPLEX STORE(NOSIG,NOSIG)
C
```

```
      COMMON /PERIPH/ IN, IOUT                                          SIS1Ø46Ø
C                                                                       SIS1Ø47Ø
C     THIS ROUTINE INVERTS THE SQUARE MATRIX                            SIS1Ø48Ø
C                                                                       SIS1Ø49Ø
      NSQ = NOSIG * NOSIG                                               SIS1Ø5ØØ
C                                                                       SIS1Ø51Ø
      DO 5 I = 1,NOSIG                                                  SIS1Ø52Ø
      DO 5 J = 1,NOSIG                                                  SIS1Ø53Ø
5     STORE(I,J) = YSIG(I,J)                                            SIS1Ø54Ø
C                                                                       SIS1Ø55Ø
      CALL FACTOR(STORE,STORE,NOSIG,IFLAG)                              SIS1Ø56Ø
C                                                                       SIS1Ø57Ø
      GO TO (2Ø,11),IFLAG                                               SIS1Ø58Ø
C                                                                       SIS1Ø59Ø
11    WRITE(IOUT,611)                                                   SIS1Ø6ØØ
611   FORMAT(1X,'THE MATRIX IS SINGULAR')                               SIS1Ø61Ø
      RETURN                                                            SIS1Ø62Ø
C                                                                       SIS1Ø63Ø
2Ø    DO 21 I = 1,NOSIG                                                 SIS1Ø64Ø
21    B(I) = CMPLX(Ø.,Ø.)                                               SIS1Ø65Ø
C                                                                       SIS1Ø66Ø
      IBEG = 1                                                          SIS1Ø67Ø
      DO 3Ø J = 1,NOSIG                                                 SIS1Ø68Ø
      B(J) = CMPLX(1.,Ø.)                                               SIS1Ø69Ø
      CALL SUBST(STORE,B,ZSIG(IBEG),NOSIG)                              SIS1Ø7ØØ
      B(J) = Ø.                                                         SIS1Ø71Ø
3Ø    IBEG = IBEG + NOSIG                                               SIS1Ø72Ø
C                                                                       SIS1Ø73Ø
      RETURN                                                            SIS1Ø74Ø
      END                                                               SIS1Ø75Ø
      SUBROUTINE FACTOR(A,W,N,IFLAG)                                    SIS1Ø76Ø
C                                                                       SIS1Ø77Ø
      COMPLEX RATIO                                                     SIS1Ø78Ø
      COMPLEX A(N,N), W(N,N)                                            SIS1Ø79Ø
      DATA TOL/1.E-7/                                                   SIS1Ø8ØØ
C                                                                       SIS1Ø81Ø
C     THIS SUBROUTINE FACTORS A GIVEN MATRIX INTO                       SIS1Ø82Ø
C     LU FORM USING GAUSSIAN ELIMINATION.                               SIS1Ø83Ø
C     SEE CONTE AND DE BOOR: 'ELEMENTARY NUMERICAL ANALYSIS'            SIS1Ø84Ø
C                                                                       SIS1Ø85Ø
      IFLAG = 1                                                         SIS1Ø86Ø
C                                                                       SIS1Ø87Ø
C     INITIALIZE W                                                      SIS1Ø88Ø
C                                                                       SIS1Ø89Ø
      DO 5 I = 1,N                                                      SIS1Ø9ØØ
      DO 5 J = 1,N                                                      SIS1Ø91Ø
5     W(I,J) = A(I,J)                                                   SIS1Ø92Ø
C                                                                       SIS1Ø93Ø
C     GAUSS ELIMINATION WITH SCALED PARTIAL PIVOTING                    SIS1Ø94Ø
C                                                                       SIS1Ø95Ø
      NMIN1 = N - 1                                                     SIS1Ø96Ø
      DO 1ØØ I = 1,NMIN1                                                SIS1Ø97Ø
      IF(CABS(W(I,I)).LT.TOL) GO TO 999                                 SIS1Ø98Ø
C                                                                       SIS1Ø99Ø
      IPLU1 = I + 1                                                     SIS11ØØØ
```

FILE: SISCAP   FORTRAN A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

```
      DO 8Ø J = IPLU1,N                                                 SIS11Ø1Ø
      W(J,I) = W(J,I) / W(I,I)                                          SIS11Ø2Ø
      RATIO = -W(J,I)                                                   SIS11Ø3Ø
      DO 4Ø K = IPLU1,N                                                 SIS11Ø4Ø
      W(J,K) = W(J,K) + RATIO * W(I,K)                                  SIS11Ø5Ø
4Ø    CONTINUE                                                          SIS11Ø6Ø
8Ø    CONTINUE                                                          SIS11Ø7Ø
1ØØ   CONTINUE                                                          SIS11Ø8Ø
      RETURN                                                            SIS11Ø9Ø
C                                                                       SIS111ØØ
999   IFLAG = 2                                                         SIS1111Ø
      RETURN                                                            SIS1112Ø
      END                                                               SIS1113Ø
      SUBROUTINE SUBST(W,B,X,N)                                         SIS1114Ø
C                                                                       SIS1115Ø
      COMPLEX SUM                                                       SIS1116Ø
      COMPLEX W(N,N), B(N), X(N)                                        SIS1117Ø
C                                                                       SIS1118Ø
C     THIS SUBROUTINE SOLVES THE MATRIX EQUATION AX=B                   SIS1119Ø
C     WITH A IN LU FORM.  SEE CONTE AND DE BOOR:                        SIS112ØØ
C     'ELEMENTARY NUMERICAL ANALYSIS'                                   SIS1121Ø
C                                                                       SIS1122Ø
      IF(N.GT.1) GO TO 1Ø                                               SIS1123Ø
      X(1) = B(1) / W(1,1)                                              SIS1124Ø
      RETURN                                                            SIS1125Ø
C                                                                       SIS1126Ø
1Ø    X(1) = B(1)                                                       SIS1127Ø
      DO 15 K = 2,N                                                     SIS1128Ø
      KM1 = K - 1                                                       SIS1129Ø
      SUM = Ø.                                                          SIS113ØØ
      DO 14 J = 1,KM1                                                   SIS1131Ø
14    SUM = W(K,J)*X(J) + SUM                                           SIS1132Ø
15    X(K) = B(K) - SUM                                                 SIS1133Ø
C                                                                       SIS1134Ø
      X(N) = X(N) / W(N,N)                                              SIS1135Ø
      K = N                                                             SIS1136Ø
      DO 2Ø NP1MK = 2,N                                                 SIS1137Ø
      KP1 = K                                                           SIS1138Ø
      K = K - 1                                                         SIS1139Ø
      SUM = Ø.                                                          SIS114ØØ
      DO 19 J = KP1,N                                                   SIS1141Ø
19    SUM = W(K,J) * X(J) + SUM                                         SIS1142Ø
2Ø    X(K) = (X(K) - SUM) / W(K,K)                                      SIS1143Ø
C                                                                       SIS1144Ø
      RETURN                                                            SIS1145Ø
      END                                                               SIS1146Ø
      SUBROUTINE FFT(NUMBER,DATA,FACTOR)                                SIS1147Ø
      INTEGER  CRPSZE,POSN,POINTR,ELMTNO,FACPTR,ABSNUM                  SIS1148Ø
      INTEGER GRPTR                                                     SIS1149Ø
      LOGICAL  INVERT                                                   SIS115ØØ
      COMPLEX  TEMP,W,FACTOR(NUMBER),DATA(NUMBER),WSCALE,TEMP1,         SIS1151Ø
     1TEMP2                                                             SIS1152Ø
     DATA  PI/3.1415926/                                                SIS1153Ø
C                                                                       SIS1154Ø
C  AUTHOR: R.G. HICKS, UNIV. OF QLD., APRIL 1977.                       SIS1155Ø
```

```
C                                                                       SIS1156Ø
C     AIM: THIS SUBROUTINE PERFORMS A FAST FOURIER TRANSFORM ON THE     SIS1157Ø
C          INPUT DATA USING THE COOLEY-TURKEY ALGORITHM                 SIS1158Ø
C          THERE MUST BE 2**NUMBER POINTS GIVEN AND THE GIVEN           SIS1159Ø
C          AND THE GIVEN POINTS ARE DESTROYED BY THE SUBROUTINE         SIS116ØØ
C          IF NUMBER IS NEGATIVE, AN INVERSE FOURIER TRANSFORM IS DONE.SIS1161Ø
C                                                                       SIS1162Ø
      INVERT = .FALSE.                                                  SIS1163Ø
      ABSNUM = NUMBER                                                   SIS1164Ø
      IF(NUMBER.GT.Ø) GO TO 1Ø                                          SIS1165Ø
      ABSNUM = -NUMBER                                                  SIS1166Ø
      INVERT = .TRUE.                                                   SIS1167Ø
1Ø    N = 2 ** ABSNUM                                                   SIS1168Ø
      ARG = 2. * PI / FLOAT(N)                                          SIS1169Ø
      TEMP = CMPLX(Ø.,ARG)                                              SIS117ØØ
      W = CEXP(TEMP)                                                    SIS1171Ø
C                                                                       SIS1172Ø
      FACTOR(1) = W                                                     SIS1173Ø
      NODIV2 = N/2                                                      SIS1174Ø
      DO 15 INDEX = 2,NODIV2                                            SIS1175Ø
15    FACTOR(INDEX) = FACTOR(INDEX - 1)*W                               SIS1176Ø
C                                                                       SIS1177Ø
      IF(INVERT) GO TO 2Ø                                               SIS1178Ø
      DO 17 POSN = 1,N                                                  SIS1179Ø
17    DATA(POSN) = CONJG(DATA(POSN))                                    SIS118ØØ
C                                                                       SIS1181Ø
2Ø    GRPSZE = N                                                        SIS1182Ø
      NOGRP = 1                                                         SIS1183Ø
      ITERNO = Ø                                                        SIS1184Ø
25    LIMIT = GRPSZE/2                                                  SIS1185Ø
      ITERNO = ITERNO + 1                                               SIS1186Ø
      POINTR = GRPSZE/2 + 1                                             SIS1187Ø
      POSN = 1                                                          SIS1188Ø
C                                                                       SIS1189Ø
      DO 6Ø GRPTR = 1, NOGRP                                            SIS119ØØ
      DO 5Ø ELMTNO = 1,LIMIT                                            SIS1191Ø
      IPOSN = POSN - 1                                                  SIS1192Ø
      CALL FACNO(IPOSN,N,ITERNO,FACPTR)                                 SIS1193Ø
      IF(FACPTR.EQ.Ø) GO TO 3Ø                                          SIS1194Ø
      WSCALE = FACTOR(FACPTR)                                           SIS1195Ø
      TEMP1 = DATA(POINTR) * WSCALE + DATA(POSN)                        SIS1196Ø
      WSCALE = -WSCALE                                                  SIS1197Ø
      TEMP2 = DATA(POINTR) * WSCALE + DATA(POSN)                        SIS1198Ø
      GO TO 4Ø                                                          SIS1199Ø
3Ø    TEMP1 = DATA(POINTR) + DATA(POSN)                                 SIS12ØØØ
      TEMP2 = DATA(POSN) - DATA(POINTR)                                 SIS12Ø1Ø
4Ø    DATA(POSN) = TEMP1                                                SIS12Ø2Ø
      DATA(POINTR) = TEMP2                                              SIS12Ø3Ø
      POSN = POSN + 1                                                   SIS12Ø4Ø
5Ø    POINTR = POINTR + 1                                               SIS12Ø5Ø
C                                                                       SIS12Ø6Ø
      POSN = POSN + GRPSZE/2                                            SIS12Ø7Ø
6Ø    POINTR = POINTR + GRPSZE/2                                        SIS12Ø8Ø
C                                                                       SIS12Ø9Ø
      GRPSZE = GRPSZE/2                                                 SIS121ØØ
      NOGRP = NOGRP * 2
      IF(GRPSZE .GT. 1) GO TO 25
C
      DO 7Ø POSN = 1,N
      IPOSN = POSN - 1
      CALL JUXTA(N,IPOSN,JUXPOS)
      JUXPOS = JUXPOS + 1
      IF(JUXPOS.LT.POSN) GO TO 7Ø
      TEMP = DATA(POSN)
      DATA(POSN) = DATA(JUXPOS)
      DATA(JUXPOS) = TEMP
7Ø    CONTINUE
C
      IF(INVERT) RETURN
      DO 8Ø POSN = 1,N
      TEMP = CONJG(DATA(POSN))
8Ø    DATA(POSN) = TEMP / N
C
      RETURN
      END
      SUBROUTINE JUXTA(N,IPOSN,JUXPOS)
      NN = N
      NUM = IPOSN
      JUXPOS = Ø
C
      DO 1Ø I=1,N
      NN = NN/2
      MEM = NUM
      NUM = NUM/2
      IREM = MEM - 2*NUM
      JUXPOS = 2*JUXPOS + IREM
      IF(NN.EQ.1) GO TO 2Ø
1Ø    CONTINUE
2Ø    RETURN
      END
      SUBROUTINE FACNO(IPOSN,N,ITERNO,FACPTR)
      INTEGER FACPTR
C
C     THIS ROUTINE IS PART OF THE BUTTERFLY CALCULATION
C     PROCEDURE IN PERFORMING A FAST FOURIER TRANSFORM
C
      FACPTR = Ø
      IDIV = N/(2**ITERNO)
      ITEMP = IPOSN
      ITEMP = ITEMP/IDIV
      MUL = N/2
      DO 1Ø I=1,ITERNO
      IPEM = ITEMP - ITEMP/2*2
      FACPTR = IREM*MUL + FACPTR
      ITEMP = ITEMP/2
      MUL = MUL/2
1Ø    CONTINUE
      RETURN
      END
      SUBROUTINE ZERO(POINTS,NUMBER)
      COMPLEX POINTS(NUMBER)                                            SIS1266Ø
      REAL MAX                                                          SIS1267Ø
      DATA TOL/1.E-5/                                                   SIS1268Ø
C                                                                       SIS1269Ø
C  AUTHOR: R.G. HICKS, UNIV. OF QLD.,  APRIL 1977.                      SIS127ØØ
C                                                                       SIS1271Ø
C     AIM: THIS SUBROUTINE AUTOMATICALLY ZEROS ANY                      SIS1272Ø
C          QUANTITIIES WHICH ARE LESS THAN TOL                          SIS1273Ø
C          TIMES THE LARGEST QUANTITY IN THE SECTION.                   SIS1274Ø
C                                                                       SIS1275Ø
C                                                                       SIS1276Ø
      MAX = CABS(POINTS(1))                                             SIS1277Ø
      DO 5 J=2,NUMBER                                                   SIS1278Ø
      IF(MAX.LT.CABS(POINTS(J))) MAX = CABS(POINTS(J))                  SIS1279Ø
5     CONTINUE                                                          SIS128ØØ
C                                                                       SIS1281Ø
      DO 1Ø J = 1,NUMBER                                                SIS1282Ø
      IF(CABS(POINTS(J)).LT.TOL*MAX) POINTS(J) = Ø.Ø                    SIS1283Ø
1Ø    CONTINUE                                                          SIS1284Ø
C                                                                       SIS1285Ø
      RETURN                                                            SIS1286Ø
      END                                                               SIS1287Ø
      SUBROUTINE THEORY(VLO,VBIAS,NMAX,IVDAT,NOIV,J,FREQ,YSIG,          SIS1288Ø
     1NOISE,TEMP,DELTAV,RN,YØ,Y1,Y1M,ZLO)                               SIS1289Ø
C                                                                       SIS129ØØ
      REAL IVDAT(NOIV), VAL(41)                                         SIS1291Ø
      COMPLEX J(NOIV), YSIG(3,3), NOISE(3,3), CVAL1, CVAL2              SIS1292Ø
      COMPLEX YØ, Y1,Y1M, ZLO, YLO                                      SIS1293Ø
      COMMON /RBOUT/ VX, VY, VJDC, VJLO, CURDC, CURPMP, RLOSS, RTEMP    SIS1294Ø
      DATA E,PI,HBAR/1.6Ø2E-19,3.1415926,1.Ø55E-34/                     SIS1295Ø
      DATA BOLTZ, TOL /1.38Ø622E-23,1.E-5/                              SIS1296Ø
C                                                                       SIS1297Ø
C     THIS SUBROUTINE SETS UP THE VALUES FOR CALCULATION                SIS1298Ø
C     OF THE SIGNAL CONVERSION AND NOISE TEMPERATURE FOR THE            SIS1299Ø
C     THREE FREQUENCY APPROXIMATION                                     SIS13ØØØ
C                                                                       SIS13Ø1Ø
      OMEGA = 2. * PI * FREQ                                            SIS13Ø2Ø
      ALPHA = E * VLO / HBAR / OMEGA                                    SIS13Ø3Ø
C                                                                       SIS13Ø4Ø
      WRITE(6,111) VBIAS, VLO, ALPHA                                    SIS13Ø5Ø
111   FORMAT(1X,'VBIAS = ',E15.8,1X,'VLO = ',E15.8,                     SIS13Ø6Ø
     11X,'ALPHA = ',E15.8)                                              SIS13Ø7Ø
C                                                                       SIS13Ø8Ø
      NMPL4 = NMAX + 1Ø                                                 SIS13Ø9Ø
      NREQ = NMPL4/2 + 1                                                SIS131ØØ
C                                                                       SIS1311Ø
      DO 1ØØ I = 1,NREQ                                                 SIS1312Ø
      N = NREQ - I                                                      SIS1313Ø
      CALL BESJ(ALPHA,N,VAL(I),TOL,IER)                                 SIS1314Ø
      IF(IER.GT.Ø) GO TO 15Ø                                            SIS1315Ø
      ICORR = NMPL4 - I + 1                                             SIS1316Ø
      VAL(ICORR) = VAL(I)                                               SIS1317Ø
      IF(N/2*2.EQ.N) GO TO 1ØØ                                          SIS1318Ø
      VAL(ICORR) = -VAL(I)                                              SIS1319Ø
1ØØ   CONTINUE                                                          SIS132ØØ
```

```
C
      GO TO 2ØØ
C
15Ø   WRITE(6,16Ø)
16Ø   FORMAT(1X,'PROBLEM IN EVALUATING THE BESSEL FUNCTION')
      STOP
C
2ØØ   CONST = HBAR * OMEGA / E
C
      GSUMØØ = Ø.
      GSUM1Ø = Ø.
      GSUMØ1 = Ø.
      GSUM11 = Ø.
      GSUM1M = Ø.
C
      BSUMØØ = Ø.
      BSUM1Ø = Ø.
      BSUMØ1 = Ø.
      BSUM11 = Ø.
      BSUM1M = Ø.
C
      HØØ = Ø.
      H1Ø = Ø.
      HØ1 = Ø.
      H11 = Ø.
      H1M = Ø.
C
      ARG = VBIAS
      ARG1 = ARG * E / HBAR
      CALL EVALJ(ARG1,CVAL1,J,NOIV,DELTAV,RN)
      CURDC = AIMAG(CVAL1)
C
      CURPMP = Ø.Ø
      CURLO1 = Ø.Ø
      CURLO2 = Ø.Ø
      CURLO3 = Ø.Ø
      CURLO4 = Ø.Ø
      CURLO5 = Ø.Ø
      CURLO6 = Ø.Ø
      CURLO7 = Ø.Ø
      CURLO8 = Ø.Ø
      CURLO9 = Ø.Ø
      CURLOØ = Ø.Ø
C
      DO 3ØØ I = 1,NMAX
C
      N = NMAX/2 - I + 1
      ARG = VBIAS + FLOAT(N) * CONST
      ARG1 = ARG * E / HBAR
      CALL EVALJ(ARG1,CVAL1,J,NOIV,DELTAV,RN)
      CALL DERIJ(ARG1,CVAL2,J,NOIV,DELTAV,RN)
      TMP1 = AIMAG(CVAL1)
      TMP2 = REAL(CVAL1)
      TMP3 = AIMAG(CVAL2)
      TMP4 = REAL(CVAL2)
```

```
C
      NPTR = NREQ - N
C
      CURPMP = VAL(NPTR)**2 * TMP1 + CURPMP
      CURLO1 = VAL(NPTR)*(VAL(NPTR-1)+VAL(NPTR+1))*TMP1 + CURLO1
      CURLO2 = VAL(NPTR)*(VAL(NPTR-1)-VAL(NPTR+1))*TMP2 + CURLO2
      CURLO3 = VAL(NPTR)*(VAL(NPTR-2)+VAL(NPTR+2))*TMP1 + CURLO3
      CURLO4 = VAL(NPTR)*(VAL(NPTR-2)-VAL(NPTR+2))*TMP2 + CURLO4
      CURLO5 = VAL(NPTR)*(VAL(NPTR-3)+VAL(NPTR+3))*TMP1 + CURLO5
      CURLO6 = VAL(NPTR)*(VAL(NPTR-3)-VAL(NPTR+3))*TMP2 + CURLO6
      CURLO7 = VAL(NPTR)*(VAL(NPTR-4)+VAL(NPTR+4))*TMP1 + CURLO7
      CURLO8 = VAL(NPTR)*(VAL(NPTR-4)-VAL(NPTR+4))*TMP2 + CURLO8
      CURLO9 = VAL(NPTR)*(VAL(NPTR-5)+VAL(NPTR+5))*TMP1 + CURLO9
      CURLO0 = VAL(NPTR)*(VAL(NPTR-5)-VAL(NPTR+5))*TMP2 + CURLO0
C
      GSUM00 = VAL(NPTR)**2 * TMP3 + GSUM00
      GSUM10 = VAL(NPTR) * (VAL(NPTR+1)+VAL(NPTR-1))*TMP3 + GSUM10
      GSUM01 = VAL(NPTR)*(VAL(NPTR+1)-VAL(NPTR-1))*TMP1+GSUM01
      GSUM11 = ((VAL(NPTR+1))**2 - (VAL(NPTR-1))**2)*TMP1 + GSUM11
      GSUM1M = VAL(NPTR)*(VAL(NPTR+2)-VAL(NPTR-2))*TMP1 + GSUM1M
C
      BSUM10 = VAL(NPTR)*(VAL(NPTR+1)-VAL(NPTR-1)) * TMP4 + BSUM10
      BSUM11 = ((VAL(NPTR+1))**2 - 2.*(VAL(NPTR))**2 +
     1 (VAL(NPTR-1))**2) * TMP2 + BSUM11
      BSUM1M = (VAL(NPTR+2)*VAL(NPTR)-2.*VAL(NPTR+1)*VAL(NPTR-1) +
     1 VAL(NPTR)*VAL(NPTR-2))*TMP2 + BSUM1M
C
      APG = ARG * E / 2. / BOLTZ / TEMP
      STO = COTH(ARG)
C
      H00 = (VAL(NPTR))**2 * STO * TMP1 + H00
      H10 = VAL(NPTR) * (VAL(NPTR+1) + VAL(NPTR-1)) * STO * TMP1
     1 + H10
      H11 = ((VAL(NPTR+1))**2 + (VAL(NPTR-1))**2) * STO * TMP1 + H11
      H1M = VAL(NPTR+1) * VAL(NPTR-1) * STO * TMP1 + H1M
300   CONTINUE
C
      GSUM10 = 0.5 * GSUM10
      GSUM01 = GSUM01 * E / HBAR / OMEGA
      GSUM11 = GSUM11 * E / HBAR / OMEGA * 0.5
      GSUM1M = GSUM1M * E / HBAR / OMEGA * 0.5
      BSUM10 = BSUM10 * 0.5
      BSUM11 = BSUM11 * E / HBAR / OMEGA / 2.
      BSUM1M = BSUM1M * E / HBAR / OMEGA / 2.
      H00 = H00 * 2. * E
      H10 = H10 * E
      H11 = H11 * E
      H1M = H1M * 2. * E
C
      GSOURC = REAL(Y1)
      BSOURC = AIMAG(Y1)
      GIMAGE = PEAL(Y1M)
      BIMAGE = AIMAG(Y1M)
      GLOAD = REAL(Y0)
      BLOAD = AIMAG(Y0)
      YLO = CMPLX(1.Ø,Ø.Ø) / ZLO                                        SIS1431Ø
      GPUMP = REAL(YLO)                                                 SIS1432Ø
      BPUMP = AIMAG(YLO)                                                SIS1433Ø
      PLO = ((CURLO1+GPUMP*VLO)**2+(CURLO2-BPUMP*VLO)**2)/8./GPUMP      SIS1434Ø
      WRITE(6,4ØØ) PLO                                                  SIS1435Ø
4ØØ   FORMAT(/1X,'LOCAL OSCILLATOR POWER = ',E15.8,' WATTS PER JN'/)    SIS1436Ø
C                                                                       SIS1437Ø
      CALL GEN3PT(GSUMØØ,GSUM1Ø,GSUMØ1,GSUM11,GSUM1M,BSUM1Ø,            SIS1438Ø
     1BSUM11,BSUM1M,HØØ,H1Ø,H11,H1M,GSOURC,GLOAD,BSOURC,GIMAGE,BIMAGE,  SIS1439Ø
     1BLOAD)                                                            SIS144ØØ
C                                                                       SIS1441Ø
      YSIG(1,1) = CMPLX(GSUM11,BSUM11)                                  SIS1442Ø
      YSIG(1,2) = CMPLX(GSUM1Ø,BSUM1Ø)                                  SIS1443Ø
      YSIG(1,3) = CMPLX(GSUM1M,BSUM1M)                                  SIS1444Ø
      YSIG(2,1) = CMPLX(GSUMØ1,Ø.)                                      SIS1445Ø
      YSIG(2,2) = CMPLX(GSUMØØ,Ø.)                                      SIS1446Ø
      YSIG(2,3) = CMPLX(GSUMØ1,Ø.)                                      SIS1447Ø
      YSIG(3,1) = CMPLX(GSUM1M,-BSUM1M)                                 SIS1448Ø
      YSIG(3,2) = CMPLX(GSUM1Ø,-BSUM1Ø)                                 SIS1449Ø
      YSIG(3,3) = CMPLX(GSUM11,-BSUM11)                                 SIS145ØØ
C      WRITE(6,2832) ((MK,ML,YSIG(MK,ML),ML=1,3),MK=1,3)                SIS1451Ø
C2832  FORMAT(1X,'MK=',I5,'ML=',I5,'YSIG=',2E15.8)                      SIS1452Ø
C                                                                       SIS1453Ø
      NOISE(1,1) = CMPLX(H11,Ø.)                                        SIS1454Ø
      NOISE(1,2) = CMPLX(H1Ø,Ø.)                                        SIS1455Ø
      NOISE(1,3) = CMPLX(H1M,Ø.)                                        SIS1456Ø
      NOISE(2,1) = CMPLX(H1Ø,Ø.)                                        SIS1457Ø
      NOISE(2,2) = CMPLX(HØØ,Ø.)                                        SIS1458Ø
      NOISE(2,3) = CMPLX(H1Ø,Ø.)                                        SIS1459Ø
      NOISE(3,1) = CMPLX(H1M,Ø.)                                        SIS146ØØ
      NOISE(3,2) = CMPLX(H1Ø,Ø.)                                        SIS1461Ø
      NOISE(3,3) = CMPLX(H11,Ø.)                                        SIS1462Ø
C                                                                       SIS1463Ø
      RETURN                                                            SIS1464Ø
      END                                                               SIS1465Ø
      SUBROUTINE DERIJ(DUM,CVAL,J,NOIV,DELTAV,RN)                       SIS1466Ø
      COMPLEX CVAL, J(NOIV), TERP5                                      SIS1467Ø
      LOGICAL FLAG                                                      SIS1468Ø
      DATA E,HBAR/1.6Ø2E-19,1.Ø55E-34/                                  SIS1469Ø
C                                                                       SIS147ØØ
C     THIS SUBROUTINE EVALUATES THE DERIVATIVE                          SIS1471Ø
C     OF J BY INTERPOLATING AMONGST A VECTOR OF J VALUES                SIS1472Ø
C                                                                       SIS1473Ø
      CONST = 1. / RN                                                   SIS1474Ø
      FLAG = .FALSE.                                                    SIS1475Ø
      ARG = DUM                                                         SIS1476Ø
      IF(ARG) 1ØØ,4ØØ,3ØØ                                               SIS1477Ø
1ØØ   FLAG = .TRUE.                                                     SIS1478Ø
      ARG = -DUM                                                        SIS1479Ø
3ØØ   PTR = ARG / DELTAV / E * HBAR                                     SIS148ØØ
      IPTR = PTR + 1.                                                   SIS1481Ø
      IF(IPTR.GE.NOIV-1) GO TO 2ØØ                                      SIS1482Ø
      RATIO = AMOD(PTR,1.)                                              SIS1483Ø
      CVAL = TERP5(J,NOIV,IPTR,RATIO,.TRUE.)/DELTAV + CMPLX(Ø.,CONST)   SIS1484Ø
      IF(FLAG) CVAL = -1. * CONJG(CVAL)                                 SIS1485Ø
      RETURN                                                            SIS1486Ø
2ØØ   CVAL = CMPLX(Ø.,CONST)                                            SIS1487Ø
      IF(FLAG) CVAL = -1. * CONJG(CVAL)                                 SIS1488Ø
      RETURN                                                            SIS1489Ø
4ØØ   CVAL = CMPLX(Ø.,Ø.)                                               SIS149ØØ
      RETURN                                                            SIS1491Ø
      END                                                               SIS1492Ø
C                                                                       SIS1493Ø
C     .............................................................    SIS1494Ø
C                                                                       SIS1495Ø
C         SUBROUTINE BESJ                                               SIS1496Ø
C                                                                       SIS1497Ø
C         PURPOSE                                                       SIS1498Ø
C            COMPUTE THE J BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDESIS1499Ø
C                                                                       SIS15ØØØ
C                                                                       SIS15Ø1Ø
C         USAGE                                                         SIS15Ø2Ø
C            CALL BESJ(X,N,BJ,D,IER)                                    SIS15Ø3Ø
C                                                                       SIS15Ø4Ø
C         DESCRIPTION OF PARAMETERS                                     SIS15Ø5Ø
C            X  -THE ARGUMENT OF THE J BESSEL FUNCTION DESIRED          SIS15Ø6Ø
C            N  -THE ORDER OF THE J BESSEL FUNCTION DESIRED             SIS15Ø7Ø
C            BJ -THE RESULTANT J BESSEL FUNCTION                        SIS15Ø8Ø
C            D  -REQUIRED ACCURACY (FRACTIONAL)                         SIS15Ø9Ø
C            IER-RESULTANT ERROR CODE WHERE                             SIS151ØØ
C               IER=Ø NØ ERROR                                          SIS1511Ø
C               IER=1 N IS NEGATIVE                                     SIS1512Ø
C               IER=2 X IS NEGATIVE OR ZERO                             SIS1513Ø
C               IER=3 REQUIRED ACCURACY NOT OBTAINED                    SIS1514Ø
C               IER=4 RANGE OF N COMPARED TO X NOT CORRECT (SEE REMARKS SIS1515Ø
C                                                                       SIS1516Ø
C         REMARKS                                                       SIS1517Ø
C            N MUST BE GREATER THAN OR EQUAL TO ZERO, BUT IT MUST BE    SIS1518Ø
C            LESS THAN                                                  SIS1519Ø
C               2Ø +1Ø*X-X** 2/3 FOR X LESS THAN OR EQUAL TO 15         SIS152ØØ
C               9Ø+X/2           FOR X GREATER THAN 15                  SIS1521Ø
C                                                                       SIS1522Ø
C         SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                 SIS1523Ø
C            NONE                                                       SIS1524Ø
C                                                                       SIS1525Ø
C         METHOD                                                        SIS1526Ø
C            RECURRENCE RELATION TECHNIQUE                              SIS1527Ø
C                                                                       SIS1528Ø
C     .............................................................    SIS1529Ø
      SUBROUTINE BESJ(XPPP,N,BJPPP,DPPP,IER)                            SIS153ØØ
      IMPLICIT REAL*8 (A-H,O-Z)                                         SIS1531Ø
      REAL BJPPP                                                        SIS1532Ø
      REAL XPPP, DPPP                                                   SIS1533Ø
      X = XPPP                                                          SIS1534Ø
      D = DPPP                                                          SIS1535Ø
C                                                                       SIS1536Ø
      BJPPP = Ø.ØEØ                                                     SIS1537Ø
      BJ=Ø.ØDØ                                                          SIS1538Ø
      IF(N)1Ø,2Ø,2Ø                                                     SIS1539Ø
1Ø    IER=1                                                             SIS154ØØ
```

```
      RETURN                                                            SIS1541Ø
2Ø    IF(X)3Ø,3ØØ,31                                                    SIS1542Ø
3Ø    IER=2                                                             SIS1543Ø
      RETURN                                                            SIS1544Ø
3ØØ   BJPPP = Ø.ØEØ                                                     SIS1545Ø
      IF(N.EQ.Ø) BJPPP = 1.ØEØ                                          SIS1546Ø
      IER = Ø                                                           SIS1547Ø
      RETURN                                                            SIS1548Ø
31    IF(X-15.DØ)32,32,34                                               SIS1549Ø
32    NTEST=2Ø.DØ+1Ø.DØ*X-X** 2.DØ/3.DØ                                 SIS155ØØ
      GOTO 36                                                           SIS1551Ø
34    NTEST=9Ø.DØ+X/2.DØ                                                SIS1552Ø
36    IF(N-NTEST)4Ø,38,38                                               SIS1553Ø
38    IER=4                                                             SIS1554Ø
      RETURN                                                            SIS1555Ø
4Ø    IER=Ø                                                             SIS1556Ø
      N1=N+1                                                            SIS1557Ø
      BPREV=Ø.ØDØ                                                       SIS1558Ø
C                                                                       SIS1559Ø
C      COMPUTE STARTING VALUE OF M                                      SIS156ØØ
C                                                                       SIS1561Ø
      IF(X-5.DØ)5Ø,6Ø,6Ø                                                SIS1562Ø
5Ø    MA=X+6.DØ                                                         SIS1563Ø
      GOTO 7Ø                                                           SIS1564Ø
6Ø    MA=1.4DØ*X+6Ø.DØ/X                                                SIS1565Ø
7Ø    MB=N+IDINT(X)/4+2                                                 SIS1566Ø
      MZERO=MAXØ(MA,MB)                                                 SIS1567Ø
C                                                                       SIS1568Ø
C      SET UPPEF LIMIT OF M                                             SIS1569Ø
C                                                                       SIS157ØØ
      MMAX=NTEST                                                        SIS1571Ø
1ØØ   DO 19Ø M=MZERO,MMAX,3                                             SIS1572Ø
C                                                                       SIS1573Ø
C      SET F(M),F(M-1)                                                  SIS1574Ø
C                                                                       SIS1575Ø
      FM1=1.ØD-28                                                       SIS1576Ø
      FM=Ø.ØDØ                                                          SIS1577Ø
      ALPHA=Ø.ØDØ                                                       SIS1578Ø
      IF(M-(M/2)*2)12Ø,11Ø,12Ø                                          SIS1579Ø
11Ø   JT=-1                                                             SIS158ØØ
      GO TO 13Ø                                                         SIS1581Ø
12Ø   JT=1                                                              SIS1582Ø
13Ø   M2=M-2                                                            SIS1583Ø
      DO 16Ø K=1,M2                                                     SIS1584Ø
      MK=M-K                                                            SIS1585Ø
      BMK=2.DØ*DFLOAT(MK)*(FM1/X)-FM                                    SIS1586Ø
      FM=FM1                                                            SIS1587Ø
      FM1=BMK                                                           SIS1588Ø
      IF(MK-N-1)15Ø,14Ø,15Ø                                             SIS1589Ø
14Ø   BJ=BMK                                                            SIS159ØØ
15Ø   JT=-JT                                                            SIS1591Ø
      S=1+JT                                                            SIS1592Ø
16Ø   ALPHA=ALPHA+BMK*S                                                 SIS1593Ø
      BMK=2.ØDØ*FM1/X-FM                                                SIS1594Ø
      IF (N)18Ø,17Ø,18Ø                                                 SIS1595Ø
```

```
170   BJ=BMK                                                            SIS15960
180   ALPHA=ALPHA+BMK                                                   SIS15970
      BJ=BJ/ALPHA                                                       SIS15980
      IF(DABS(BJ-BPREV)-DABS(D*BJ))200,200,190                          SIS15990
190   BPREV=BJ                                                          SIS16000
      IER=3                                                             SIS16010
200   BJPPP = BJ                                                        SIS16020
      RETURN                                                            SIS16030
      END                                                               SIS16040
      FUNCTION COTH(ARG)                                                SIS16050
      REAL NUM, NEG                                                     SIS16060
      NEG = -ARG                                                        SIS16070
      NUM = EXP(ARG) + EXP(NEG)                                         SIS16080
      DEN = EXP(ARG) - EXP(NEG)                                         SIS16090
      IF(ABS(DEN).LT.1.E-5) GO TO 10                                    SIS16100
      COTH = NUM / DEN                                                  SIS16110
      RETURN                                                            SIS16120
10    COTH = 1.E10                                                      SIS16130
      RETURN                                                            SIS16140
      END                                                               SIS16150
      SUBROUTINE GEN3PT(GSUM00,GSUM10,GSUM01,GSUM11,GSUM1M,             SIS16160
     1BSUM10,BSUM11,BSUM1M,H00,H10,H11,H1M,GSOURC,GLOAD,BSOURC,         SIS16170
     2GIMAGE,BIMAGE,BLOAD)                                              SIS16180
      COMMON /RBOUT/ VDC, VLO, VJDC, VJLO, CURDC, CURPMP, LOSS, CREAL   SIS16190
C                                                                       SIS16200
      REAL L0, LOSS                                                     SIS16210
      COMPLEX LAM01, CTM                                                SIS16220
      COMPLEX Z01, LAM01M                                               SIS16230
      DATA TOL/1.E-5/, BOLTZ/1.380622E-23/                              SIS16240
C                                                                       SIS16250
C     THIS SUBROUTINE CALCULATES THE SMALL SIGNAL AND                   SIS16260
C     NOISE PARAMETERS USING THE VALUES THAT HAVE BEEN                  SIS16270
C     SET UP USING SUBROUTINE 'THEORY'                                  SIS16280
C                                                                       SIS16290
      L0 = 2. * GSUM10 / GSUM01                                         SIS16300
      CSI = (GSUM11 - GSUM1M)/(GSUM11 + GSUM1M)                         SIS16310
      GAMMA = BSUM1M / (GSUM11 + GSUM1M)                                SIS16320
      ETA = 2. * GSUM01 * GSUM10 / GSUM00 / (GSUM11 + GSUM1M)           SIS16330
      GL = GLOAD / GSUM00                                               SIS16340
      BETA = BSUM10 / GSUM10                                            SIS16350
      BI = BIMAGE / (GSUM11 + GSUM1M)                                   SIS16360
      GA = (GSOURC + GIMAGE) / 2.0 / (GSUM11 + GSUM1M)                  SIS16370
      GD = (GSOURC - GIMAGE) / 2.0 / (GSUM11 + GSUM1M)                  SIS16380
      BA = (BSUM11 + (BSOURC + BIMAGE) / 2.0) / (GSUM11 + GSUM1M)       SIS16390
      BD = (BSOURC - BIMAGE) / 2.0 / (GSUM11 + GSUM1M)                  SIS16400
      BL = BLOAD / GSUM00                                               SIS16410
C                                                                       SIS16420
      TERM1 = ((CSI + GA) * (1.0 + GA) + (BA**2 - GAMMA**2) -           SIS16430
     1GD**2 - BD**2) * (1.0 + GL + BL)                                  SIS16440
      TERM2 = (BD * (1.0 + CSI + 2.0*GA) - 2.0*GD*BA) *                 SIS16450
     1(1.0 + GL + BL)                                                   SIS16460
      TERM3 = -1.0 * ETA * (CSI + GA + BETA*BA - BETA*GAMMA)            SIS16470
      TERM4 = -1.0 * ETA * (BD - BETA * GD)                             SIS16480
      TERM5 = GSUM00 * (GSUM11 + GSUM1M) **2                            SIS16490
      TERM6 = -1.0 * GSUM01 * (GSUM11+GSUM1M) * (CSI + GA - GD)         SIS16500
      TERM7 = -1.Ø * GSUMØ1 * (GSUM11+GSUM1M) * (GAMMA + BD - BA)
      ZR1 = TERM5 * (TERM1 + TERM3)
      ZI1 = TERM5 * (TERM2 + TERM4)
      ZR2 = TERM6
      ZI2 = TERM7
      ZØ1 = CMPLX(ZR2,ZI2) / CMPLX(ZR1,ZI1)
      LOSS = 1./4./GSOURC/GLOAD/CABS(ZØ1)**2
      LOSS = 1Ø. * ALOG1Ø(LOSS)
C
      TERM1 = -GSUMØ1 * (GSUM11+GSUM1M) * (CSI+GA-GD)
      TERM2 = -GSUMØ1 * (GSUM11+GSUM1M) * (GAMMA+BD-BA)
      TERM3 = (GSUM11 + GSUM1M) ** 2
      TERM4 = (CSI+GA) * (1.Ø+GA) + BA**2 - GAMMA**2 - GD**2 - BD**2
      TERM5 = BD * (1.Ø+CSI+2.Ø*GA) - 2.Ø*GD*BA
      TERM6 = TERM3 * TERM4
      TERM7 = TERM3 * TERM5
      LAMØ1 = CMPLX(TERM1,TERM2) / CMPLX(TERM6, TERM7)
      TERM8 = -GSUMØ1 * (GSUM11+GSUM1M) * (CSI+GA+GD)
      TERM9 = -GSUMØ1 * (GSUM11+GSUM1M) * (BA+BD-GAMMA)
      LAMØ1M = CMPLX(TERM8,TERM9) / CMPLX(TERM6,TERM7)
      CTM = (CABS(LAMØ1M)**2 + CABS(LAMØ1)**2) * H11
     1 + (LAMØ1M+CONJG(LAMØ1M)+CONJG(LAMØ1)+LAMØ1) * H1Ø
     2 + HØØ + (LAMØ1M*CONJG(LAMØ1)+LAMØ1*CONJG(LAMØ1M))*H1M
      CTM = 1./4./GSOURC/CABS(LAMØ1)**2*CTM/BOLTZ
      CREAL = REAL(CTM)
      CIMAG = AIMAG(CTM)
      IF(CIMAG.GT.TOL*CREAL) GO TO 1ØØØ
C
      WRITE(6,8Ø) LOSS, CREAL
8Ø    FORMAT(1X,'CONVERSION LOSS (USB) = ',E15.8,' DB '/
     11X,'SSB SIS MIXER NOISE TEMPERATURE (USB) = ',E15.8,' K')
      RETURN
C
1ØØØ  WRITE(6,1ØØ1)
1ØØ1  FORMAT(1X,'PROBLEM DEVELOPED IN TEMPERATURE COMPUTATION')
      STOP
      END
      COMPLEX FUNCTION TERP5(J,NOIV,IPTR,RATIO,FLAG)
      COMPLEX J(NOIV)
      COMPLEX STORE(4)
      LOGICAL FLAG
      DATA N/3/
C
C     THIS COMPLEX VALUED FUNCTION PERFORMS AN INTERPOLATION
C     AMONGST THE ARRAY OF J VALUES.
C     IF FLAG IS TRUE, AN INTERPOLATION OF DERIVATIVES IS DONE
C     IF FLAG IS FALSE, AN INTERPOLATION OF THE FUNCTION
C     J IS DONE.
C     THE INTERPOLATING FUNCTION USED IS A+BX+CEXP(DX)
C     AND THE ARRAY OF J VALUES MUST BE EVENLY SPACED
C
      NP1 = N + 1
      MID = (N + 1) / 2
      DO 1 K = 1,NP1
      KK = IPTR - (MID - 1) + (K-1)
      IF(KK.LE.Ø) GO TO 2
      STORE(K) = J(KK)
      GO TO 1
2     KK = 2 - KK
      STORE(K) = CONJG(J(KK))
1     CONTINUE
C
      YM1 = REAL(STORE(1))
      YØ = REAL(STORE(2))
      Y1 = REAL(STORE(3))
      Y2 = REAL(STORE(4))
C
      CURV1 = Y2 - 2.Ø*Y1 + YØ
      CURV2 = Y1 - 2.Ø*YØ + YM1
      IF(CURV1*CURV2 .LT. Ø.Ø) GO TO 47Ø
      IF(ABS(CURV1) .LE  Ø.Ø .OR. ABS(CURV2).LE.Ø.Ø) GO TO 47Ø
      D = ALOG(CURV1/CURV2)
      IF(D.EQ.Ø) GO TO 47Ø
      C = CURV1 / ((EXP(D) - 1.Ø) ** 2)
      A = YØ - C
      B = Y1 - A - C*EXP(D)
      VALR = A + B*RATIO + C*EXP(D*RATIO)
      IF(FLAG) VALR = B + C*D*EXP(D*RATIO)
      GO TO 4ØØ
C
47Ø   VALR = RATIO * (Y1 - YØ) + YØ
      IF(FLAG) VALR = Y1 - YØ
C
4ØØ   YM1 = AIMAG(STORE(1))
      YØ = AIMAG(STORE(2))
      Y1 = AIMAG(STORE(3))
      Y2 = AIMAG(STORE(4))
C
      CURV1 = Y2 - 2.Ø*Y1 + YØ
      CURV2 = Y1 - 2.Ø*YØ + YM1
      IF(CURV1*CURV2.LT.Ø.Ø) GO TO 57Ø
      IF(ABS(CURV1).LE.Ø.Ø .OR. ABS(CURV2).LE.Ø.Ø) GO TO 57Ø
      D = ALOG(CURV1/CURV2)
      IF(D.EQ.Ø) GO TO 57Ø
      C = CURV1 / ((EXP(D) - 1.Ø) ** 2)
      A = YØ - C
      B = Y1 - A - C*EXP(D)
      VALI = A + B*RATIO + C*EXP(D*RATIO)
      IF(FLAG) VALI = B + C*D*EXP(RATIO*D)
      GO TO 5ØØ
C
57Ø   VALI = RATIO * (Y1 - YØ) + YØ
      IF(FLAG) VALI = Y1 - YØ
C
5ØØ   TERP5 = CMPLX(VALR,VALI)
      RETURN
      END
      SUBROUTINE MRT(VOLT,CUR,NOPNTS,IBASE1,CHI,NOCHI,
     1RN,FACTOR.FREQ,ZIMPED,VSOURC,OLDVOL,FLAG,ZØ,NOPER)
C
```

```
      REAL VOLT(NOPNTS), CUR(NOPNTS), CHI(NOCHI), OLDVOL(NOPNTS),
     1ESOURC(128)
      COMPLEX FACTOR(NOPNTS), ZIMPED(NOPNTS), VSOURC(NOPNTS),
     1ELEFT(128), ERIGHT(128), RHO(128)
      LOGICAL FLAG
C
C     THIS SUBROUTINE CONTROLS THE MULTIPLE REFLECTION
C     LARGE SIGNAL ANALYSIS PROCEDURE
C
      INTER = -IBASE1
      NODIV2 = NOPNTS / 2
C
C     INITIALIZE THE LEFT AND RIGHT PROPAGATING WAVES
C
      DO 1Ø I = 1,NODIV2
      ICORR = NOPNTS - I + 2
      ELEFT(I) = CMPLX(Ø.Ø,Ø.Ø)
      ERIGHT(I) = 2.Ø * (ZØ/(ZIMPED(I)+ZØ)*VSOURC(I))
      IF(I.EQ.1) GO TO 1Ø
      ELEFT(ICORR) = CONJG(ELEFT(I))
      ERIGHT(ICORR) = CONJG(ERIGHT(I))
1Ø    CONTINUE
C
      ELEFT(NODIV2 + 1) = CMPLX(Ø.Ø,Ø.Ø)
      ERIGHT(NODIV2 + 1) = CMPLX(Ø.Ø,Ø.Ø)
C
      CALL FFT(INTER,ERIGHT,FACTOR)
C
      DO 15 I = 1,NOPNTS
      ESOURC(I) = ERIGHT(I) / ZØ
      OLDVOL(I) = Ø.Ø
15    CONTINUE
C
C     CALCULATE THE REFLECTION COEFFICIENT OF THE EMBEDDING
C     NETWORK AT EACH FREQUENCY OF INTEREST
C
      DO 17 I = 1,NODIV2
      RHO(I) = (ZIMPED(I) - ZØ) / (ZIMPED(I) + ZØ)
      ICORR = NOPNTS - I + 2
      IF(I.EQ.1) GO TO 17
      RHO(ICORR) = CONJG(RHO(I))
17    CONTINUE
      RHO(NODIV2 + 1) = CMPLX(Ø.Ø,Ø.Ø)
C
      WRITE(6,6Ø4)
6Ø4   FORMAT(/1X,'GENERAL NONLINEAR ANALYSIS USING MULTIPLE',
     1' REFLECTION TECHNIQUE'/1X,'BEGINNING THE ITERATION PHASE')
C
      DO 1ØØØ II = 1,NOPER
C
      WRITE(6,4ØØ) II
4ØØ   FORMAT(/1X,'ITERATION NUMBER = ',I5)
C
C     CALCULATE THE TUNNEL JUNCTION VOLTAGE AND CURRENT WAVEFORMS
C
```

```
      CALL SISCHA(ESOURC,NOPNTS,ZØ,VOLT,CUR,CHI,NOCHI,RN,FREQ,IBASE1,   SIS1816Ø
     1NOPER,FACTOR)                                                     SIS1817Ø
C                                                                       SIS1818Ø
C     DETERMINES THE REVISED LEFT PROPAGATING WAVE                      SIS1819Ø
C                                                                       SIS182ØØ
      DO 2Ø I = 1,NOPNTS                                                SIS1821Ø
      ELEFT(I) = (VOLT(I) - CUR(I) * ZØ) * Ø.5                          SIS1822Ø
2Ø    CONTINUE                                                          SIS1823Ø
C                                                                       SIS1824Ø
      CALL FFT(IBASE1,ELEFT,FACTOR)                                     SIS1825Ø
C                                                                       SIS1826Ø
C     DETERMINE THE REVISED RIGHT PROPAGATING WAVE                      SIS1827Ø
C                                                                       SIS1828Ø
      DO 3Ø I = 1,NODIV2                                                SIS1829Ø
      ICORR = NOPNTS - I + 2                                            SIS183ØØ
      ERIGHT(I) = (ZØ/(ZIMPED(I)+ZØ)*VSOURC(I) + RHO(I)*ELEFT(I))*2.Ø   SIS1831Ø
      IF(I.EQ.1) GO TO 3Ø                                               SIS1832Ø
      ERIGHT(ICORR) = CONJG(ERIGHT(I))                                  SIS1833Ø
3Ø    CONTINUE                                                          SIS1834Ø
      ERIGHT(NODIV2 + 1) = CMPLX(Ø.Ø,Ø.Ø)                               SIS1835Ø
C                                                                       SIS1836Ø
      CALL FFT(INTER,ERIGHT,FACTOR)                                     SIS1837Ø
      DO 4Ø I = 1,NOPNTS                                                SIS1838Ø
      ESOURC(I) = ERIGHT(I) / ZØ                                        SIS1839Ø
4Ø    CONTINUE                                                          SIS184ØØ
C                                                                       SIS1841Ø
C     SEE IF A SOLUTION HAS BEEN OBTAINED                               SIS1842Ø
C                                                                       SIS1843Ø
      CALL COMPAR(VOLT,OLDVOL,NOPNTS,FLAG)                              SIS1844Ø
C                                                                       SIS1845Ø
      DO 5ØØ JJ = 1,NOPNTS                                              SIS1846Ø
5ØØ   OLDVOL(JJ) = VOLT(JJ)                                             SIS1847Ø
C                                                                       SIS1848Ø
      IF(.NOT.FLAG) GO TO 2ØØØ                                          SIS1849Ø
C                                                                       SIS185ØØ
1ØØØ  CONTINUE                                                          SIS1851Ø
C                                                                       SIS1852Ø
      WRITE(6,15ØØ) NOPER                                               SIS1853Ø
15ØØ  FORMAT(/8X,'LARGE SIGNAL ANALYSIS DID NOT CONVERGE IN ',I5,'ITNS')SIS1854Ø
      RETURN                                                            SIS1855Ø
C                                                                       SIS1856Ø
2ØØØ  WRITE(6,2ØØ2) II                                                  SIS1857Ø
2ØØ2  FORMAT(/1X,'LARGE SIGNAL ANALYSIS COMPLETED IN ',I5,' ITNS'/)     SIS1858Ø
      RETURN                                                            SIS1859Ø
      END                                                               SIS186ØØ
      SUBROUTINE SISCHA(ESOURC,NOPNTS,RSOURC,VOLT,CUR,CHI,NOCHI,        SIS1861Ø
     1RN,FREQ,IBASE,NOPER,FACTOR)                                       SIS1862Ø
C                                                                       SIS1863Ø
      REAL ESOURC(NOPNTS), VOLT(NOPNTS), CUR(NOPNTS)                    SIS1864Ø
      REAL CHI(NOCHI)                                                   SIS1865Ø
      COMPLEX FACTOR(NOPNTS)                                            SIS1866Ø
      COMMON /RBOUT/ VDC,VLO,VJDC,VJLO,CURDC,CURPMP,RLOSS,RTEMP         SIS1867Ø
C                                                                       SIS1868Ø
      REAL ZTERM(512), PHASE(512), VOLT2(512)                           SIS1869Ø
      COMPLEX UTERM(512), CCUR(128)                                     SIS187ØØ
```

```
C
C     SET UP PARAMETERS FOR THE CALL TO SUBROUTINE
C     SIS TO DETERMINE THE JUNCTION VOLTAGE AND CURRENT
C     WAVEFORMS
C
      HDEL = 1.Ø / FREQ / FLOAT(NOPNTS)
C
      IADJST = Ø
      CALL SIS(CHI,VOLT2,NOPNTS,HDEL,ZTERM,UTERM,
     1RN,NOCHI,NOPER,PHASE,NOSTOP,IADJST,RSOURC,ESOURC,ISTO)
C
      DO 1ØØØ I =1,NOPNTS
      IPNT = (ISTO - 1) * NOPNTS + I - IADJST
      VOLT(I) = VOLT2(IPNT)
      CUR(I) = ESOURC(I) - VOLT(I) / RSOURC
      CCUR(I) = CUR(I)
1ØØØ  CONTINUE
C
      CALL FFT(IBASE,CCUR,FACTOR)
      CURPMP = CCUR(1)
C
      RETURN
      END
      SUBROUTINE SIS(CHI,VOLT,NOPNTS,HDEL,ZTERM,UTERM,
     1RN,NOCHI,NOPER,PHASE,NOSTOP,IADJST,RSOURC,ESOURC,ISTO)
C
      INTEGER R, RR, VCNT
      REAL ZTERM(NOCHI), VOLT(NOCHI), CHI(NOCHI), PHASE(NOCHI)
      REAL ESOURC(NOPNTS)
      COMPLEX UTERM(NOCHI), SUM, CDUM, CONST, CSUM, CTEMP
      LOGICAL FLAG
      DATA HBAR, E/1.Ø55E-34,1.6Ø2E-19/
      DATA TOL, ITYPE/1.E-5,1/
      DATA NHIGH/1ØØ/
      DATA VTHRSH/Ø ØE-3/
C
C     DETERMINE THE SIS TUNNEL JUNCTION CURRENT AND
C     VOLTAGE WAVEFORM COMPONENTS
C
      RNN = 1./(1./RN + 1./RSOURC)
      NOSTOP = NOCHI
      UTERM(1) = CMPLX(1.,Ø.)
      VOLT(1) = Ø.
      ZTERM(1) = Ø.
      PHASE(1) = Ø.
      CONST = CMPLX(Ø.,-1.) * E / HBAR * HDEL
C
      DO 1ØØØ I = 1,NOPER
      ISTO = I
      VCNT = Ø
C
      DO 5ØØ RR = 1,NOPNTS
C
      N = NOPNTS * (I-1) + RR
C
```

```
FILE: SISCAP   FORTRAN  A   NASA/GISS CONVERSATIONAL MONITOR SYSTEM SP3Ø2

C     IF NECESSARY, SHIFT ALL VALUES LEFT IN THE STORAGE VECTORS        SIS1926Ø
C                                                                       SIS1927Ø
      IF(N.LT.NOCHI) GO TO 658                                          SIS1928Ø
      DO 657 NPRO = 2,NOCHI                                             SIS1929Ø
      VOLT(NPRO-1) = VOLT(NPRO)                                         SIS193ØØ
      PHASE(NPRO-1) = PHASE(NPRO)                                       SIS1931Ø
      UTERM(NPRO-1) = UTERM(NPRO)                                       SIS1932Ø
      ZTERM(NPRO-1) = ZTERM(NPRO)                                       SIS1933Ø
657   CONTINUE                                                          SIS1934Ø
C                                                                       SIS1935Ø
      IADJST = IADJST + 1                                               SIS1936Ø
C                                                                       SIS1937Ø
C     INITIALLY ESTIMATE THE NEXT VOLTAGE POINT BY                      SIS1938Ø
C     ASSUMING A CONSTANT VOLTAGE                                       SIS1939Ø
C                                                                       SIS194ØØ
658   VOLT(N-IADJST+1) = VOLT(N-IADJST)                                 SIS1941Ø
      PHASE(N-IADJST+1) = PHASE(N-IADJST) + AIMAG(CONST) *              SIS1942Ø
     1VOLT(N-IADJST+1)                                                  SIS1943Ø
      CALL FIXPHA(PHASE(N-IADJST+1))                                    SIS1944Ø
      CTEMP = CMPLX(Ø.,PHASE(N-IADJST+1))                               SIS1945Ø
      UTERM(N-IADJST+1) = CEXP(CTEMP)                                   SIS1946Ø
      VPREV = VOLT(N-IADJST+1)                                          SIS1947Ø
      VMID1 = VOLT(N-IADJST+1)                                          SIS1948Ø
      VMID2 = VOLT(N-IADJST+1)                                          SIS1949Ø
C                                                                       SIS195ØØ
      CSUM = CMPLX(Ø.,Ø.)                                               SIS1951Ø
C                                                                       SIS1952Ø
C     ITERATE TO DETERMINE THE CORRECT VOLTAGE                          SIS1953Ø
C     AT THE NEXT POINT IN TIME                                         SIS1954Ø
C                                                                       SIS1955Ø
      DO 425 MM = 1,NHIGH                                               SIS1956Ø
C                                                                       SIS1957Ø
      SUM = CMPLX(Ø.,Ø.)                                                SIS1958Ø
C                                                                       SIS1959Ø
      IF(MM.NE.1 .OR. N.EQ.1) GO TO 243                                 SIS196ØØ
C                                                                       SIS1961Ø
C     PERFORM THE REQUIRED CONVOLUTION                                  SIS1962Ø
C                                                                       SIS1963Ø
      NMIN1 = N - 1                                                     SIS1964Ø
      ITPRY = IADJST + 1                                                SIS1965Ø
      DO 1ØØ R = ITPRY,NMIN1                                            SIS1966Ø
      CSUM = (CHI(N-R+1)*UTERM(R-IADJST+1) + CHI(N-R+2)*UTERM(R-IADJST))SIS1967Ø
     1/2. + CSUM                                                        SIS1968Ø
1ØØ   CONTINUE                                                          SIS1969Ø
C                                                                       SIS197ØØ
243   SUM = (CHI(1)*UTERM(N-IADJST+1) + CHI(2)*UTERM(N-IADJST))/2.      SIS1971Ø
     1+ CSUM                                                            SIS1972Ø
C                                                                       SIS1973Ø
      IF(MM.NE.1) GO TO 1Ø1                                             SIS1974Ø
      YF = YFUNCT(N,HDEL,CHI,NOCHI)                                     SIS1975Ø
      N2 = (N+1) - (N/NOPNTS*NOPNTS)                                    SIS1976Ø
      CC = ESOURC(N2)                                                   SIS1977Ø
1Ø1   CDUM = CONJG(UTERM(N-IADJST+1)) * SUM * HDEL +                    SIS1978Ø
     1CONJG(UTERM(N-IADJST+1)) * YF                                     SIS1979Ø
      ZTERM(N-IADJST+1) = -CC + AIMAG(CDUM)                             SIS198ØØ
```

```
C
C      VTST = ((-1./CAP*(ZTERM(N-IADJST+1)+ZTERM(N-IADJST))/2.
C     1-1./2./CAP/RNN*VOLT(N-IADJST)) *
C     1HDEL + VOLT(N-IADJST))/(1.+HDEL/2./RNN/CAP)
C
      VTST = -1.Ø * RNN * ZTERM(N-IADJST+1)
C
      ITS2 = N - IADJST + 1
      IF(ABS(VTST-VOLT(ITS2)).LT.TOL*ABS(VOLT(ITS2)))
     1GO TO 45Ø
      IF(MM/2*2.EQ.MM) GO TO 421
C
423   VOLT(N-IADJST+1) = (VTST + VOLT(N-IADJST+1))/2.
      PHASE(N-IADJST+1) = PHASE(N-IADJST) + AIMAG(CONST) *
     1(VOLT(N-IADJST+1)+VOLT(N-IADJST))/2.
      CALL FIXPHA(PHASE(N-IADJST+1))
      CTEMP = CMPLX(Ø.,PHASE(N-IADJST+1))
      UTERM(N-IADJST+1) = CEXP(CTEMP)
      GO TO 422
C
421   IF(ABS(VMID2-VPREV) .LT. TOL*ABS(VMID2)) GO TO 423
      VG = (VTST - VMID1) / (VMID2 - VPREV)
      IF(ABS(1.-VG) .LT. TOL) GO TO 423
      VOLT(N-IADJST+1) = (VMID1 - VG*VPREV) / (1. - VG)
      PHASE(N-IADJST+1) = PHASE(N-IADJST) + AIMAG(CONST) *
     1(VOLT(N-IADJST+1) + VOLT(N-IADJST))/2.
      CALL FIXPHA(PHASE(N-IADJST+1))
      CTEMP = CMPLX(Ø.,PHASE(N-IADJST+1))
      UTERM(N-IADJST+1) = CEXP(CTEMP)
C
422   VMID1 = VTST
      VPREV = VMID2
      VMID2 = VOLT(N-IADJST+1)
425   CONTINUE
C
      WRITE(6,234)
234   FORMAT(1X,'******PROBLEM: ITERATION LOOP 1 EXPIRED')
C
C     STOP
C
45Ø   ITS1Ø = N - IADJST + 1
      IF(VOLT(ITS1Ø) .LT. VTHRSH) VCNT = VCNT + 1
C
      IF(RR.EQ.NOPNTS) GO TO 6ØØ
5ØØ   CONTINUE
C
C     A WATCH IS KEPT ON HOW MANY VOLTAGE POINTS IN
C     THE WAVEFORMS ARE NEGATIVE VOLTAGES
C
6ØØ   IF(VCNT .EQ. Ø) GO TO 62Ø
      WRITE(6,6Ø1) VCNT, I
6Ø1   FORMAT(1X,'SIS:VOLTAGE THRESHOLD COUNTER = ',I5,'ITN NO = ',I5)
C
62Ø   IF(I.LE.ITYPE) GO TO 1ØØØ
      CALL COMP2(VOLT,I,NOPNTS,FLAG,ITYPE,NOCHI.IADJST)
      IF(FLAG) GO TO 2000                                               SIS20360
1000  CONTINUE                                                          SIS20370
C                                                                       SIS20380
      WRITE(6,233)                                                      SIS20390
233   FORMAT(1X,'******PROBLEM: ITERATION LOOP 2 EXPIRED')              SIS20400
      NOSTOP = N + 1                                                    SIS20410
      RETURN                                                            SIS20420
C                                                                       SIS20430
2000  WRITE(6,655) ITYPE, ISTO                                          SIS20440
655   FORMAT(1X,'SIS:PERIODICITY OBTAINED WITH ITYPE = ',I5,            SIS20450
     1/1X,'SIS:NO. OF ITERATIONS = ',I5)                                SIS20460
      NOSTOP = N + 1                                                    SIS20470
      RETURN                                                            SIS20480
      END                                                               SIS20490
      FUNCTION YFUNCT(N,HDEL,CHI,NOCHI)                                 SIS20500
C                                                                       SIS20510
      REAL CHI(NOCHI)                                                   SIS20520
C                                                                       SIS20530
C     THIS FORMS PART OF THE CALCULATION OF THE CONVOLUTION             SIS20540
C     FORMULA IN THE SIS EQUATIONS                                      SIS20550
C                                                                       SIS20560
      YFUNCT = 0.0                                                      SIS20570
      IF(N.GE.NOCHI) RETURN                                             SIS20580
C                                                                       SIS20590
      NPLUS1 = N + 1                                                    SIS20600
      NMIN1 = NOCHI - 1                                                 SIS20610
      DO 20 I = NPLUS1,NMIN1                                            SIS20620
      YFUNCT = YFUNCT + (CHI(I)+CHI(I+1))/2.0*HDEL                      SIS20630
20    CCNTINUE                                                          SIS20640
      RETURN                                                            SIS20650
C                                                                       SIS20660
      END                                                               SIS20670
      SUBROUTINE COMP2(VOLT,I,NOPNTS,FLAG,ITYPE,NOCHI,IADJST)           SIS20680
      LOGICAL FLAG                                                      SIS20690
      REAL VOLT(NOCHI)                                                  SIS20700
      DATA TOL, EPS/1.E-4,1.E-1/                                        SIS20710
C                                                                       SIS20720
C     THIS SUBROUTINE COMPARES THE TWO MOST RECENT                      SIS20730
C     SUCCESSIVE VOLTAGE WAVEFORMS IN THE SIS WAVEFORM                  SIS20740
C     ITERATION PROCESS                                                 SIS20750
C                                                                       SIS20760
      EMAX = 0.0                                                        SIS20770
      IPNT = 0                                                          SIS20780
      FLAG = .TRUE.                                                     SIS20790
      IF(I.LE.ITYPE) GO TO 1000                                         SIS20800
C                                                                       SIS20810
      CMAX = 0.                                                         SIS20820
      DO 100 M = 1,NOPNTS                                               SIS20830
      MM = (I - 1) * NOPNTS + M - IADJST                                SIS20840
      IF(ABS(VOLT(MM)) .GT. CMAX) CMAX = ABS(VOLT(MM))                  SIS20850
100   CONTINUE                                                          SIS20860
C                                                                       SIS20870
      IF(CMAX .LT. 1.E-12) RETURN                                       SIS20880
C                                                                       SIS20890
      DO 500 J = 1,NOPNTS                                               SIS20900
      IPOINT = (I-1) * NOPNTS + J - IADJST                              SIS2Ø91Ø
      ISEEK = IPOINT - NOPNTS * ITYPE                                   SIS2Ø92Ø
      IF(ISEEK.LE.Ø) GO TO 5ØØ                                          SIS2Ø93Ø
      IF(ABS(VOLT(IPOINT)).LT.EPS*CMAX) GO TO 5ØØ                       SIS2Ø94Ø
      ERROR = (VOLT(IPOINT)-VOLT(ISEEK))/VOLT(IPOINT)                   SIS2Ø95Ø
      ERROR = ABS(ERROR)                                                SIS2Ø96Ø
      IF(ERROR .LT. EMAX) GO TO 4ØØ                                     SIS2Ø97Ø
      EMAX = ERROR                                                      SIS2Ø98Ø
      IPNT = J                                                          SIS2Ø99Ø
4ØØ   IF(ERROR .LT. TOL) GO TO 5ØØ                                      SIS21ØØØ
      FLAG = .FALSE.                                                    SIS21Ø1Ø
5ØØ   CONTINUE                                                          SIS21Ø2Ø
      RETURN                                                            SIS21Ø3Ø
C                                                                       SIS21Ø4Ø
1ØØØ  FLAG = .FALSE.                                                    SIS21Ø5Ø
      RETURN                                                            SIS21Ø6Ø
      END                                                               SIS21Ø7Ø
      SUBROUTINE FIXPHA(X)                                              SIS21Ø8Ø
      DATA PI/3.1415926/                                                SIS21Ø9Ø
C                                                                       SIS211Ø2
C     THIS PERFORMS A MODULO FUNCTION ON THE PHASE                      SIS2111Ø
C     CALCULATIONS IN THE SIS EQUATIONS                                 SIS2112Ø
C                                                                       SIS2113Ø
      IF(ABS(X).LT.2.Ø*PI) RETURN                                       SIS2114Ø
      NOREV = X / 2.Ø / PI                                              SIS2115Ø
      X = X - 2.Ø * PI * FLOAT(NOREV)                                   SIS2116Ø
      RETURN                                                            SIS2117Ø
      END                                                               SIS2118Ø
```

ACKNOWLEDGEMENTS

The authors wish to thank S.-K. Pan and P. Siegel for many helpful discussions. R.G. Hicks gratefully acknowledges the financial support of the National Research Council Fellowship program.

REFERENCES

[1] J.R. Tucker, "Quantum limited detection in tunnel junction mixers", IEEE Jnl. Quantum Electronics, vol. 15, Nov. 1979, pp. 1234-1258.

[2] M.J. Feldman, S.-K. Pan, A.R. Kerr and A. Davidson, "SIS mixer analysis using a scale model", IEEE Trans. Magn., Vol. MAG-19, pp. 494-497, May 1983.

[3] P.H. Siegel and A.R. Kerr, "A user orientated computer program for the analysis of microwave mixers, and a study of the effects of the series inductance and diode conductance on the performance of some simple mixers", NASA Technical Memorandum 80324, Goddard Space Flight Center, Institute for Space Studies, Greenbelt, Maryland 20771, July 1979.

[4] J.R. Tucker and M.J. Feldman, "Quantum detection at millimeter wavelengths", Rev. Mod. Phys. Vol. 57, pp. 1055-1113, Oct., 1985.

[5] W.R. McGrath, A.V. Raisanen and P.L. Richards, "Precise noise measurements of superconducting quasiparticle mixers", presented at 1984 Applied Superconductivity Conference, San Diego, CA, Sept. 9-13, 1984.

[6] H.C. Torrey and C.A. Whitmer, "Crystal Rectifiers", MIT Radiation Lab. Series, vol. 15, New York, McGraw Hill, 1948.

[7] D.A. Fleri and L.D. Cohen, "Nonlinear analysis of the Schottky-barrier mixer diode", IEEE Trans. on Microwave Theory and Tech., vol. MTT-21, no. 1, Jan. 1973, pp. 39-43.

[8] S. Egami, "Nonlinear, linear analysis and computer aided design of resistive mixers", IEEE Trans. on Microwave Theory and Tech., vol. MTT-22, no. 3, March 1974, pp. 270-275.

[9] W.K. Gwarek, "Nonlinear analysis of microwave mixers", M.S. Thesis, M.I.T., Cambridge, Mass., 1974.

[10] A.R. Kerr, "A technique for determining the local oscillator waveforms in a microwave mixer", IEEE Trans. on Microwave Theory and Tech., vol. MTT-23, no. 10, Oct. 1975, pp. 828-831.

[11] R.G. Hicks and P.J. Khan, "Numerical analysis of nonlinear solid-state device excitation in microwave circuits", IEEE Trans. on Microwave Theory and Tech., vol. MTT-30, no. 3, March 1982, pp. 251-259.

[12] R.G. Hicks and P.J. Khan, "Numerical technique for determining pumped nonlinear device waveforms", Electronics Letters, vol. 16, no. 10, May 1980, pp. 375-376.

[13] D.N. Held and A.R. Kerr, "Conversion loss and noise of microwave and millimeter-wave mixers: Part I - Theory", IEEE Trans. Micro. Theory Tech., Vol. MTT-26, pp.49-55, Feb. 1978.

[14] P.H. Siegel, A.R. Kerr and W. Huang, "Topics in the optimization of millimeter-wave mixers", NASA Technical Paper 2287, Goddard Institute for Space Studies, New York, NY, March 1984.

[15] A.A.M. Saleh, "Theory of resistive mixers", MIT Press, Cambridge, Mass., 1971.

[16] R.N. Bracewell, "The Fourier transform and its applications", New York, McGraw-Hill, 1978, pp. 267-272.

[17] J.R. Tucker, "The quantum response of nonlinear tunnel junctions as detectors and mixers", Reviews in Infrared and Millimeter Waves, Vol. I, pp. 1-46, 1983.

[18] M.J. Feldman and S. Rudner, "Mixing with SIS arrays", Reviews in Infrared and Millimeter Waves, Vol. I, pp. 47-75, 1983.

[19] C.M. Caves, "Quantum limits on noise in linear amplifiers", Phys. Rev. D, Vol. 26, pp. 1817-1838, Oct. 1982.

[20] C. Dragone, "Analysis of thermal and shot noise in pumped resistive diodes", Bell System Tech. Journal, vol. 47, pp. 1883-1902, 1968.

[21] A. Uhlir, "Shot noise in p-n junction frequency converters", Bell. Syst. Tech. Journal, vol. 37, no. 4, July 1958, pp. 951-988.

[22] R.I. Gayley, "Josephson tunnel junction time-dependence calculations based on the Bardeen-Cooper-Schrieffer theory", J. Appl. Phys., vol. 52, no. 3, March 1981, pp. 1411-1416.

[23] R.G. Hicks and M.J. Feldman, "Chaos in SIS quasiparticle tunneling", Proc. Eighth Intnl. Conf. on Infrared and Millimeter Waves, p. T6.6, Dec. 1983.

[24] Lead alloy two-junction array PNBS-W2-C27-36 fabricated by Peter Timbie of Princeton University at the NBS Boulder facility.

1 Report No NASA TM-86145	2 Government Accession No	3 Recipient's Catalog No
4 Title and Subtitle A GENERAL NUMERICAL ANALYSIS PROGRAM FOR THE SUPERCONDUCTING QUASIPARTICLE MIXER		5 Report Date July 1986 6 Performing Organization Code 640
7 Author(s) Ross G. Hicks, Marc J. Feldman, and Anthony R. Kerr		8 Performing Organization Report No 10 Work Unit No
9 Performing Organization Name and Address NASA/Goddard Institute for Space Studies New York, NY 10025		11 Contract or Grant No 13 Type of Report and Period Covered Technical Memorandum
12 Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14 Sponsoring Agency Code

15 Supplementary Notes

16 Abstract

This report describes user oriented computer program SISCAP (SIS Computer Analysis Program) for analysing SIS mixers. The program allows arbitrary impedance terminations to be specified at all LO harmonics and sideband frequencies. It is therefore able to treat a much more general class of SIS mixers than the widely used three-frequency analysis, for which the harmonics are assumed to be short-circuited. An additional program, GETCHI, provides the necessary input data to program SISCAP.

The program SISCAP performs a nonlinear analysis to determine the SIS junction voltage waveform produced by the local oscillator. The quantum theory of mixing is used in its most general form, treating the large signal properties of the mixer in the time domain. A small signal linear analysis is then used to find the conversion loss and port impedances. The noise analysis, includes thermal noise from the termination resistances and shot noise from the periodic LO current. Quantum noise is not considered.

Many aspects of the program have been adequately verified and found accurate. However, this project was terminated before the program was extensively employed, and it is possible that errors may occur in various untested parameter ranges.

An example is given, which forms part of some initial investigations into the effect of decreasing the SIS junction capacitance on the performace of a simple SIS mixer circuit. Early indications are that, for this circuit, a high $\omega_p R_N C_J$ product is desirable.

17 Key Words (Suggested by Author(s)) SIS Junctions Input Impedance Quantum Noise	18 Distribution Statement Unclassified - Unlimited Subject Category 61

19 Security Classif (of this report)	20 Security Classif (of this page)	21 No of Pages	22 Price
Unclassified	Unclassified	80	A05

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1986

End of Document